





David Ahl's Basic Computer Adventures

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
16011 N.E. 36th Way, Box 97017
Redmond, Washington 98073-9717

Copyright © 1986 by David Ahl
All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging in Publication Data
Ahl, David H.

Ahl's BASIC computer adventures.

Bibliography: p.

1. Computer adventure games. 2. BASIC (Computer program language).

I. Title. II. Title: BASIC computer adventures.

GV1469.22.A35 1986 794'.8 86-18095

ISBN 0-914845-92-6

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 MLML 8 9 0 9 8 7 6

Distributed to the book trade in the United States by Harper & Row.

Distributed to the book trade in Canada by General Publishing Company, Ltd.

Distributed to the book trade outside the United States and Canada by Penguin Books Ltd.

Penguin Books Ltd., Harmondsworth, Middlesex, England

Penguin Books Australia Ltd., Ringwood, Victoria, Australia

Penguin Books N.Z. Ltd., 182-190 Wairau Road, Auckland 10, New Zealand

British Cataloging in Publication Data available

Apple® is a registered trademark of, and Macintosh™ is a trademark licensed to Apple Computer, Incorporated.

Atari® is a registered trademark of Atari, Incorporated.

Commodore™ is a trademark of Commodore Business Machines, Incorporated.

Microsoft® is a registered trademark of Microsoft Corporation.

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



DEDICATION

To Amelia Earhart, Francis Parkman, Marco Polo and the many other bold and resolute adventurers who proved that no frontier is beyond the reach of man.

PREFACE

Although one can trace the origins of digital computers to the Analytical Engine of Charles Babbage or the punched card tabulating machines of Herman Hollerith, the real beginning of the computer age was in the late 1930s and early 1940s, sparked in large part by various aspects of World War II. In Britain, Alan Turing's group at Bletchley Park was trying to build a computer for code-breaking purposes. In Germany, Konrad Zuse built a relay-type machine that could store 64 floating point numbers for rocketry calculations. In the U.S., several developments were proceeding in parallel, but by far the most influential was the work of John Mauchly and J. Presper Eckert at the University of Pennsylvania on a machine which was eventually christened ENIAC (Electronic Numerical Integrator and Computer).

The Purpose of ENIAC was to integrate ballistic equations for gunnery tables. Such tables show the required angle of elevation of a gun for various target distances, shell weights, and wind speeds. A significant accomplishment at the dedication of ENIAC in February 1946 was the computation of the trajectory of a 16" naval shell in less than real time. Since World War II had ended by the time ENIAC was operational, there was no longer the urgent need for the firing tables that motivated its design. Thus, after it was moved to Aberdeen Proving Ground, it was put to use in atomic energy calculations, cosmic ray studies, thermal ignition research, wind tunnel design, and weather prediction as well as for its original purpose of computing ballistic tables for the Army and Air Force.

Unfortunately, today we tend to get a somewhat distorted view of the history of digital computers. Books written by writers in the computer field tend to focus on the early technology (vacuum tubes, mercury delay lines, paper tape), capacity, speed, and occasionally, people. On the other hand, general history books tend to focus on data processing and financial applications, and ignore the incredibly important role of the military in perhaps the most momentous development of the 20th century.

Equally salient is the fact that virtually all of the early applications of digital computers were simulations—of an artillery weapon, a nuclear reaction, a wind tunnel, the weather. It was not until some years later that computers were put to work in data processing, financial applications, and communications.

MY EARLY INVOLVEMENT

In 1957, just 11 years after the first digital computer was demonstrated, in my sophomore year at Cornell University, I got my first taste of computing. By then, of course, computers were being widely used for data processing and related applications, but I was far more fascinated with their ability to simulate the real world. Summers during college, I worked at Grumman Aircraft refurbishing war-worn airplanes, testing antenna configurations, and writing programs with the computer group. I'll never forget the countless hours I spent in the computer room as our project team tried to simulate the actions of an airplane pilot in various situations.

The following summer, I was in a group charged with writing programs to simulate the movements of an early satellite, the orbiting astronomical observatory (OAO). The experience I gained working on these simulations put me in a good position to undertake my senior project at Cornell, a massive program (in ALGOL) that simulated the acoustics of a concert hall. Given the size, shape, seating configuration, wall covering, and ceiling material of the hall, the program calculated the acoustical properties of various seating locations. Today, such a program seems almost primitive, but in 1960 it was a major accomplishment.

A year later, in the MBA program at Carnegie-Mellon University, I had the opportunity to work with the team that wrote the first management game, a simulation of three companies competing in the detergent market. Since

then, as my career has progressed through positions at Management Science Associates, Digital Equipment Corp., AT&T, and Creative Computing, I have had fewer and fewer opportunities to program. However, over the years, in the back of my mind I tucked away ideas for future programs.

Thus, when Ziff-Davis decided to fold Creative Computing magazine, it was with a sense of anticipation that I remembered some of these program ideas. Claudette Moore, formerly an editor at Creative Computing, now at Microsoft Press, and I were chatting one day and I mentioned the idea of doing a series of travel simulations. She, with some enthusiasm, asked me to put together a proposal for a book. I did, and this is it. Now, I can only hope that you, the reader, learn as much from the stories and have as much fun taking the simulated journeys as I had researching and writing them.

In closing let me thank Betsy Staples for her enormous help and fanatic attention to detail in editing the manuscript, Claudette Moore for encouraging me to write the book in the first place, and Jody Gilbert, David Rygmyr, and all the other folks at Microsoft Press for seeing the project through to completion.

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



INTRODUCTION

WRITING A SIMULATION

When writing a large and complex program, the first question always is where to start? The answer is, start with a pad of paper and DON'T turn on the computer.

Writing a computer simulation of some real or imaginary system is not especially difficult. The much bigger job is gathering the data and learning as much as possible about the system so you can define the relationships among its various facets. Once you understand the whole system, you must then break it down into the smallest possible pieces, analyze each piece in depth, determine how to model it, and combine the pieces back into the whole.

In developing the algorithms and writing the programs for each piece, you must determine the most accurate and efficient way to proceed. Is it an exact process that can be represented by a mathematical formula? Is the process continuous or discontinuous? Is it ever interrupted abruptly? Are there random elements involved?

An enormous amount of time must be spent testing each piece. Does the simulation behave in a sensible way? What happens at various extremes? These same questions must be asked again and again as all of the pieces are combined into the whole system.

Like most real-world systems, most of the simulations in this book are a combination of well-defined processes and random elements. For example, Amelia Earhart's airplane flies and consumes fuel according to well-defined physical principles. On the other hand, in Southeast Asia in July, she has a 30% chance of running into a torrential rainstorm. If it occurs, the problems she faces will be quite different from those on a sunny day.

There are five distinct parts to each of the simulations: an introduction to the game, historical background, program notes, program variables, and the program listings. I encourage you to read The Story (historical background) before taking each computerized journey. By doing so, you will have a much better understanding of the original journey and you will make better decisions as you go along.

Later, if you want to see why the program works the way it does, examine how a routine operates, or even modify the program, the program notes and the list of variables will serve as your guide. You may also find in these programs many interesting and useful routines that you can use in programs of your own.

The ten simulations in this book are rather diverse. Some require many decisions on the part of the player, and some require relatively few decisions. In some, they are easy to make, but random hazards make it difficult to finish the journey. In others, the decisions are more difficult, but the journey itself is easier.

The programs are arranged in chronological order ranging from Marco Polo (1271) to Voyage to Neptune (2100). I suggest you play them in the order in which they appear. Following is a brief preview of what to expect in each simulation. Other aspects of the programs are summarized in the following table:

The Game	The Date	The Mode of Transportation	The Duration
Marco Polo	1271	Camel, foot	3 years
Westward Ho!	1847	Wagon, oxen	6 months
The Longest Automobile Race	1908	Automobile	6 months
The Orient Express	1923	Train	4 days
Amelia Earhart	1937	Airplane	1 month
Tour de France	present	Bicycle	3 weeks
Appalachian Trail	present	Foot	5 months
Subway Scavenger	present	Subway train	8 hours

Hong Kong Hustle	1997	Mixed (bus, tram, ferry, train, subway)	1 day
Voyage to Neptune	2100	Spaceship	6 years

SUMMARY OF THE ADVENTURE SIMULATIONS

- Marco Polo simulates a three-year journey over the Silk Road in 1271. It is easy to play once you get the hang of using jewels in place of money. However, it is a challenge to complete the journey in three years and have enough jewels remaining for trading.
- Westward Ho! simulates a six-month journey over the Oregon Trail in 1850. It uses the same framework as Marco Polo but, because you have fewer resources, it is slightly more difficult to finish the trip.
- The Longest Automobile Race simulates the 1908 auto race from New York to Paris (east to west). In those days, the cars were unreliable, the roads were nothing more than cart tracks, and, as always, the weather was unpredictable. It is not too difficult to reach Paris, but quite difficult to win the race.
- The Orient Express simulates a mystery case aboard the famous train in 1923. You will almost always complete the trip, but the mystery is extremely difficult to solve.
- Amelia Earhart simulates the around-the-world flight attempt of the famous aviatrix in 1937. If you load enough fuel and pay attention to maintenance and repairs, you will have no trouble reaching Southeast Asia. From there on, it is tough going.
- Tour de France simulates the famous three-week bicycle race. Along with intelligent decisions, this game requires manual dexterity on the keyboard. It is reasonably easy to finish, but quite difficult to win.
- Appalachian Trail simulates the trek of a hiker walking from Georgia to Maine. Preparation is half the key to completing the hike, but sensible decisions along the Trail about eating, walking speed, and walking hours are important also.
- Subway Scavenger simulates the rounds of a messenger using the New York subway system. During the first few plays of the game, you must develop a map of the subway system. Even with the help of a map, you will find this an extremely challenging game to finish.
- Hong Kong Hustle simulates the Tai Pan collecting bags of gold from various locations around Hong Kong. Using the same framework as Subway Scavenger, you must first develop a map, then go for the gold.
- Voyage to Neptune simulates an imaginary journey to Neptune in the year 2100. It is a game of trading and tradeoffs. It is not difficult to get to Neptune; it is very difficult to do so in six years or less and have enough fuel left for the return trip.

Following the ten simulations is a section on machine requirements and converting the programs to other versions of BASIC. Be warned that converting programs as long as these is a major undertaking.

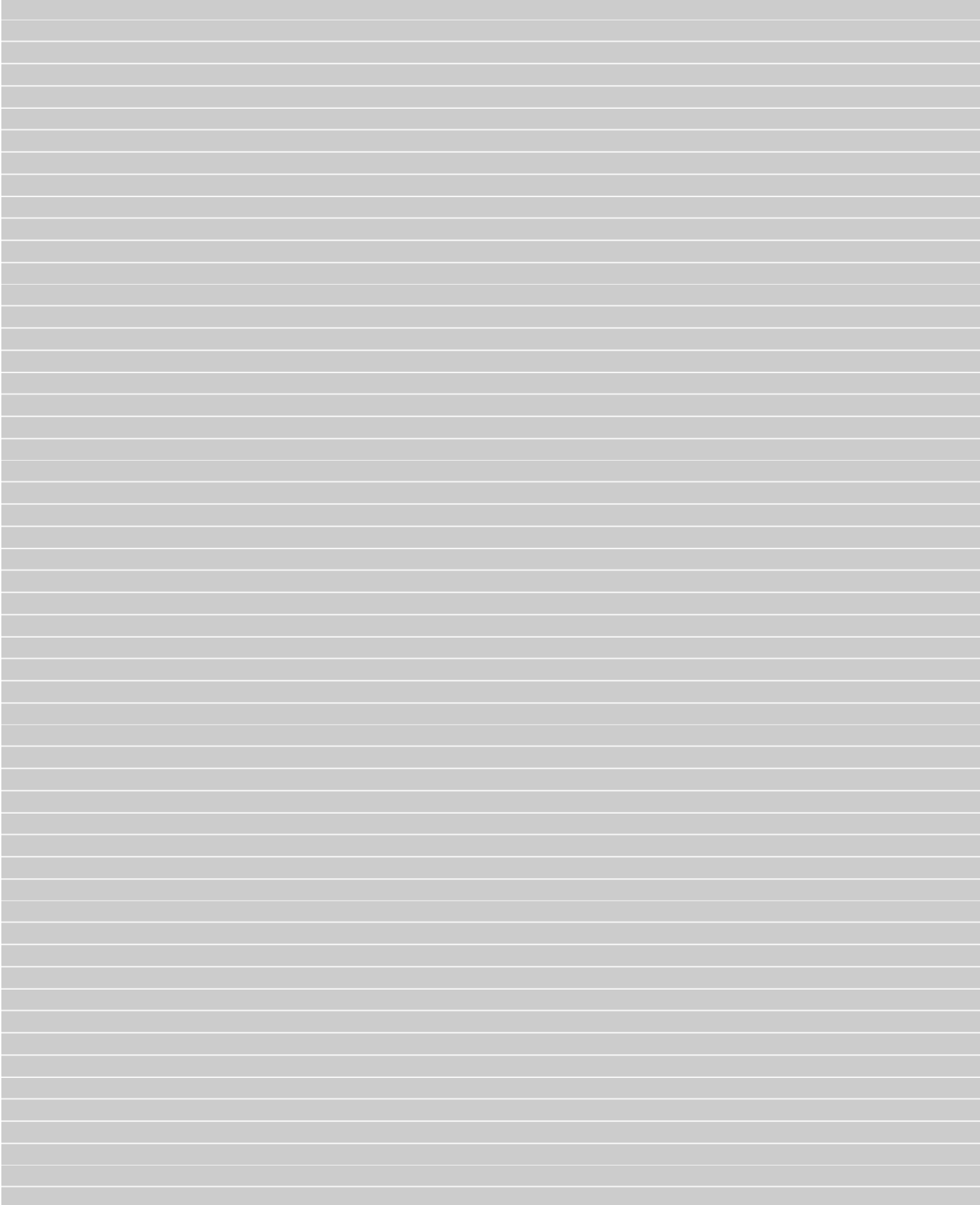
I hope you have fun trekking through this volume. Good luck on your computerized journeys!

COMPANION DISK TO BASIC COMPUTER ADVENTURES

To save you hours of typing in the programs—and the additional hours of finding those inevitable typos—David Ahl has created a special program disk for *BASIC Computer Adventures*. The disk contains all 10 programs included in the book, plus loading instructions for the IBM PC, PCjr, and PC-compatible computers.

~~This disk is only available directly from David Ahl. To order, send \$20.00 (plus \$1.00 per disk for domestic postage and handling charges, \$2.00 for foreign orders) to: David Ahl, 12 Indian Head Road, Morristown, NJ 07960. Payment must be in U.S. funds. You may pay by check or money order (payable to David Ahl). Please allow 4 weeks for delivery.~~

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)





THE GAME

In Marco Polo you are the leader of a group of merchants leaving on a business trip from Venice, Italy to Shang-tu, China. After sailing to Armenia at the east end of the Mediterranean Sea, you set out by camel on a 6000-mile trek across Asia to the court of the Great Kublai Khan in Shang-tu near the Pacific Ocean. To get your party safely to Shang-tu, you must know the rules of the Silk Road; the more important ones are noted here:

- At the beginning of the game, you are asked to rank your shooting ability. Since the program cannot really test your prowess with a crossbow, shooting ability is measured by how fast you can type a random shooting sound. A rating of 1 describes a reasonably good typist, while 4 should be reserved for those who use no more than two fingers. The game will be more fun if you rank yourself 1 or 2.
- Approximately every two months during your trip, you come upon a village that has a market where you may replenish your provisions. As prices vary widely from place to place, it is best to maintain a modest stock of goods so you are not forced to buy at high prices.
- Deciding how well to eat during each two-month period involves some tradeoffs. Eating better allows you to walk longer and cover more ground; you are also less susceptible to disease. However, food costs money (jewels), and your camels have a limited carrying capacity. If you run out of food on the trip, you can always eat a camel (assuming you have one left).
- Balm and unguents are used for treating wounds. If you run out, you face a much higher risk of getting a fatal infection.
- As a merchant, you are not a skilled hunter. However, occasionally you may be offered an opportunity to hunt. Count it a blessing if you get food in this way, but remember that your crossbow is the only weapon

you have with which to drive off bandits. Hence, you should always keep a small supply of arrows in reserve.

Many hazards and surprises await you along the road to Shang-tu, so stay alert and keep your wits about you. The Polos completed the land journey in approximately 36 months. In the game, this means arriving at Shang-tu in March 1274. If you make only the best decisions and encounter no delays, it is possible to complete the computer journey in 24 months, but a more realistic goal is to complete the trip in the same 36 months it took the Polos. Can you do it?

THE STORY

Today, a traveler flying into Venice, Italy lands at Marco Polo Airport. At the Doge's Palace in Piazza San Marco, one can view magnificent tapestries, porcelain, and jade carvings from China—the rewards of being at the western terminus of the "Silk Road" to the Far East, a route first traveled by Marco Polo in 1271.

Yet upon his return from his now famous journey to the Far East, Marco Polo's tales of strange people and far off places met with disbelief. For hundreds of years, to call something a "Marco Polo" was to label it a tall tale or even an outright falsehood. Even after they were published, his stories were thought to be largely fictional accounts. Indeed, there is little to indicate that a single one of Marco Polo's contemporaries believed much of his story. And, on his deathbed his friends pleaded with him, for the peace of his soul, to retract some of the incredible statements made in his book. Instead, he refused and is said to have replied, "I have not told half of what I saw."

As his accounts were set down many years before the development of printing, the volumes were copied by hand, and variations in wording and numerous embellishments crept into the work.

In all, more than 100 different manuscripts were produced—some in Italian, some in Latin, and some in French. The earliest printed edition is dated 1559, and an English translation did not appear until 1818.

Not until the late 1800's did scholars attempt to piece together a truly original edition, as reports from later travelers and explorers began to indicate that the majority of Marco Polo's accounts were accurate and unexaggerated. Unfortunately, not one edition treating the entire work as a travel narrative has ever appeared, nor has the story ever been translated into contemporary English. Thus, to this day, for most Americans, the myths and the realities remain intertwined.

Marco Polo was born in Venice in 1254, son of Niccolò, one of the great merchants and noblemen of the city. An uncle, Maffeo, worked directly with his father, and together the team of brothers traveled to many distant lands.

Niccolò Polo and his brother Maffeo made their first great journey east in 1260. They visited their third brother in Constantinople and from there set out on a trading trip along the Tigris River to the great city of Bokhara in the

Persian Empire (today, the city of Bukhara is in south central Russia). There they met an ambassador of the Great Khan (Supreme Lord), Kublai, son of the conquering Genghis Khan, who lived at the eastern extremity of the continent in Shang-tu (today, the inconsequential town of Shangdu about 200 miles northwest of Beijing, China).

Kublai Khan had never seen a native of Italy and requested an audience with the two brothers. Convinced that they had no choice in the matter, the brothers made an incredible year-long journey across Tibet and Mongolia to the eastern-most part of the Empire of the Great Khan (Cathay, or China).

As the first Europeans to set foot in the court of the Great Khan, they were entertained with feasts and plied with extravagant gifts. Kublai Khan questioned the brothers at length and became convinced that his Empire could benefit greatly from European learning (although, it should be mentioned, the Khanates were in many respects better governed and more civilized than much of Europe at that time). Consequently, Kublai Khan asked the brothers to relay to the Pope a request for 100 men of learning who could be stationed throughout his extensive empire to disseminate the best of Western culture.

Although furnished with escorts, provisions, and everything necessary for their return journey, the Polos remained subject to the hazards of travel—extreme cold, snow, floods, deserts, and diseases—and it was three years before they reached the seaport of Laissus in Armenia and set sail for Venice.

Back in Italy, they found that Pope Clement IV had just died.

Two years passed before they could relay the Khan's request to the new Pope, Gregory X, who, instead of furnishing 100 men of learning, dispatched two friars of the Order of Preachers to accompany the Polos on their return trip. Having heard accounts of warring tribes along the route, the friars feared for their lives and, after just a few days' journey, turned back. This was not the only time this happened; as H.G. Wells in *The Outline of History* reports, "This abortive mission was only one of a number of attempts to communicate, and always they were feeble and feeble-spirited attempts, with nothing of the conquering fire of the earlier Christian missions."

Manuel Komroff in *The Travels of Marco Polo* goes further. "A hundred cultured men living in China at this time and returning home at various periods would have changed the course of human events. Europe was just awakening from a long, barbaric sleep, while China was already cultured in many fields. Marco Polo came to exchange merchandise, while 100 cultured men would have returned to exchange ideas. It is the traffic of ideas that is of greater profit to humanity."

On the second journey, the two Polo brothers decided to bring Niccolò's 16-year-old son, Marco, along. Marco was, first and foremost, a merchant, and much of his journals discuss trade, finance, risk, and profit. He also had an eye for nature and described many varieties of birds, trees, and other plants and animals. But beyond the realms of commerce and nature he was largely without vision and simply reported what he saw in a matter-of-fact style with little analysis of the underlying whys and wherefores.

The second journey of the Polos resembled the first—the main difference being the eyewitness account provided by young Marco's notes. As mentioned, these notes are not in the form of a travel narrative, but rather a

description of things and places. Moreover, in setting down his account, Marco rearranged his notes to tell of a country (or city) and its immediate neighbors, thus making it difficult to define the actual route taken.

However, by comparing Marco's accounts with other historical information, excavations, and legends, historians have accurately reconstructed the route of this second legendary journey. Rather than starting in Constantinople, the second journey started in the port of Laissus in Lower Armenia (today, near Adana in south central Turkey). From there the travelers headed northeast along the Euphrates River and then turned southward along the Tigris River to Babylon (Baghdad) and continued on south to the Persian Gulf.

From there they continued south to Hormuz, where the caravan turned almost due north to cross the Dasht-e Lut and Dasht-e Kavir desert regions of Persia to Herat (then in the Khanate of Persia, today in Afghanistan). Next they followed a difficult trek across the mountains of Afghanistan, skirting north of Kashmir to Kashgir, the capital city of the Khanate of Chaghadai (today, Kashi, China).

Continuing in the mountains, the Polos then descended and crossed the narrowest part of the desert of Lop, which took a month. As Marco described, "During these days the journey is invariably over either sandy plains or barren mountains. In this tract, neither beasts or birds are met with, because there is no kind of food for them." He also described "excessive troubles and dangers that must unavoidably be encountered" such as mirages, malevolent spirits, eerie noises, and the danger of losing the path. This is one of the only places in which Marco Polo discussed the dangers of the route, so it must be supposed that they made a great impression on him. To this day this bone-strewn and barren waste has been crossed by very few travelers, and it remains one of the most desolate regions of the world.

The caravan then continued into the province of Tanguth in the Khanate of the Great Khan along what is today the border of Tibet (Xizang) and Sinkiang (Xinjiang). They continued generally eastward, veering off to the north before reaching Xian, the legendary eastern terminus of the Silk Road. The northern route followed the Yellow River for 550 miles, but unfortunately it also obliged the Polos to cross a portion of the Gobi Desert to reach Shang-tu. Marco did not dwell as long on the Gobi Desert as he did on the Lop, although he did mention that one must "lay in provisions for at least 40 days because that space of time is employed in traversing the desert, where there is not any appearance of a dwelling, nor are there any inhabitants."

Finally, after traveling for three and a half years, the Polos arrived in the court of the Great Khan and bowed low before the emperor. In place of 100 learned men, they had with them a few letters from the new Pope, a little sacred oil from the Holy Land, and a few items to trade. By this time Marco was 21, the year, 1275.

Kublai Khan took a liking to Marco Polo, who at once applied himself to learning the written and spoken languages of the country. The Emperor, seeing that the young man was both clever and tactful, began to send him on public missions to other parts of the empire.

Marco Polo had observed that the Khan was often bored by the dry reports of his administrators but enjoyed hearing about the manners and oddities of people in other regions. Thus Marco started to keep small notebooks of strange facts that were likely to amuse and interest Kublai Khan. It was from these notebooks that Marco eventually transcribed the account of his travels back in Italy.

The Polos prospered in the court of Kublai Khan, and the Khan became very attached to them. Although they wanted to return to Italy, the Khan apparently felt that in a small way they were serving in place of the 100 men he had requested and declined to let them go.

However, 20 years later the Khan of Persia lost his favorite wife and asked Kublai Khan to send him another from the same Mongol tribe from which she had come. The Polos, who were expert navigators, proposed to the Khan that they be allowed to pilot the ships that would carry the party to Persia. Reluctantly, the Khan consented.

The Polos exchanged all their acquired possessions for jewels and set sail on a long and dangerous two-year voyage through the South China Sea and the Indian Ocean to Persia.

A year later, after having left the spectacular court of Kublai Khan, Niccolò, Maffeo, and Marco Polo arrived in their old home, Venice. Their clothes were tattered and foreign, their faces reflected the ravages of travel, and they had practically forgotten their native tongue. They had long been thought dead, and the distant relatives occupying their house refused to admit them after their absence of 26 years.

They finally succeeded in convincing their kindred that they were not imposters, and a great feast was arranged. All their old friends and relatives were invited. The Polos dressed in new velvet and damask garments for the meal, but when the table had been cleared and all the servants asked to leave, Marco Polo produced the coarse,

shabby costumes they had worn on their arrival. Then, taking sharp knives, they ripped the seams and let fall to the table quantities of rubies, sapphires, diamonds, pearls, and other jewels. The guests were amazed and dumbfounded, the story spread, and the Polos became the most illustrious merchants of Venice.

Because the Polos were merchants, they immediately set themselves up in business and again began to trade. At the time, there were fierce rivalries among the great Italian merchant cities of Venice, Pisa, and Genoa. These rivalries had reached the point of open warfare, and most merchant families maintained one or more war galleys to protect their harbors and trading ships from both pirates and truculent rivals.

In a major battle, the Venetian and Genoese fleets met on September 7, 1298, just three years after the Polos' return from the Far East. In the battle, the Genoese captured the entire Venetian fleet and took 7000 Venetians, including Marco Polo, prisoner. Most were released in exchange for ransom, but the Genoese refused to release Marco Polo.

Thus, in a Genoese jail, Marco Polo dictated the notes of his travels to a fellow prisoner, Rusticien, a scribe from Pisa, and they were set down on parchment. Within a year, the merchant war between Venice and Genoa was over, Marco Polo was released, and the world got its first, disbelieving glimpse of the strange and fascinating land of Asia.

BIBLIOGRAPHY

Fairbank, J. K., Reischauer, E. O., and Craig, A. M. *East Asia: Tradition and Transformation*. Boston: Houghton Mifflin, 1973.

Komroff, Manuel. *The Travels of Marco Polo*. New York: The Modern Library, 1953.

Wells, H. G. *The Outline of History: A Plain History of Life and Mankind*. Boston: Doubleday, 1971.

THE PROGRAM

The Marco Polo program consists of a very short main program that simply calls a series of ten major subroutines. Ten shorter subroutines perform frequently used operations such as checking for a yes/no answer or centering a printed line. The program uses the same framework as the Westward Ho! program which includes extensive program notes; hence these notes describe only those sections that are unique. Both the subroutine to deal with the purchase of initial supplies (Lines 720–880) and the one to deal with bartering for supplies along the way (Lines 1190–1380) require the player to specify quantities of six items. These quantities must be in a range, the lower limit of which is usually zero and the upper limit of which is defined by the number of jewels the player has. Hence, before each INPUT statement, the upper limit is set:

```
A2 = (INT(JL / RN))
```

in which A2 is the upper limit, JL is the number of jewels, and RN is the cost of the item. The cost is a random function which varies over a small range. After setting the limit, an INPUT statement gets the answer, following which a subroutine (Lines 3790–3820), which checks to see if the answer is within the acceptable range, is called. If not, a message saying, "That is too few" or "That is too many," is printed and the input request is stated again.

Consider the expression in Line 1580:

```
FR = INT(.5 + 10 * (F - FE)) / 10
```

The purpose of this expression is to calculate a fractional value (to 1/10) of the food reserve (FR), which is food (F) less food eaten (FE). This function is necessary because the computer may not calculate fractional values absolutely correctly (2/5 may come out 0.40000001, for example) and we do not want to print these extraneous digits or, worse yet, compound the error as the program proceeds. Because the integer (INT) function actually truncates the decimal places, it is necessary to add 0.5 to ten times the number to retain the correct decimal value. Try calculating the value of FR with and without the 0.5 for values of F–FE equal to 3.299999 and 3.300001.

Another expression frequently used in the program is found at Line 2160:

```
IF FC = 1 THEN X$ = "" ELSE X$ = "S"
```

This is used when printing the reference to an item, in this case sacks of food, that could have a value of zero, one, or more. When printing, we want the singular or plural form of the item to correspond to the numeric value, i.e., "I sack" and "2 sacks." Thus, the PRINT statement uses the string X\$ as follows:

```
PRINT "You'll have to sell" FC "sack" X$ "of food"
```

Notice the spacing. When a numeric value (FC) is printed, it is automatically preceded and followed by a space; thus, we enclose no spaces in the quotes on either side of a numeric value. However, a string (X\$) has no automatic spaces and because we want a space following the word "sack" or "sacks," we leave a space inside the quotes following X\$.

Compare the method of reading the probabilities of events occurring in this program (Lines 3570–3580) with that in Westward Ho! (Lines 3710–3720). The effect is the same, but the Marco Polo approach makes it easier to change the probabilities.

```
3570 FOR I = 1 TO 14 : READ A : EPT = EPT + A : EP(I) = EPT : NEXT I
3580 DATA 6, 4, 4, 6, 6, 6, 6, 4, 4, 1, 6, 8, 18, 10
```

The probability of the first event occurring is arbitrarily set to 6. The next event is only two-thirds as likely to happen as the first; it is assigned a value of four. The 13th event is three times as likely to happen as the first; it is assigned a value of 18.

The cumulative value of these events is then read into the array EP. EP(1) = 6, EP(2) = 6 + 4 or 10, EP(3) = 14, and so on.

The function in Line 1950 determines which event occurs:

```
RN = INT(EPT * RND(1))
```

This function selects a value between 1 and EPT. If RN is between 1 and 6, event 1 occurs; between 6 and 10, event 2; between 10 and 14, event 3; and so on.

The function in Westward Ho! does the same thing but it requires you to calculate in advance the value of EPT and all the values of EP(n). Consequently, it is not nearly as easy to see the relationship between various events. Nor is it as easy to change things around, because changing one value at the beginning of the list will affect all later values.

The shooting subroutines in Marco Polo (Lines 3620–3700) and Westward Ho! are also very similar. First, a random shooting word (SPLAT, TWACK, etc.) is selected in Line 3630 which you are directed to type. Then, the current time (in seconds) is read from the real-time clock of the computer in Line 3640; this is the starting time. An INPUT statement in Line 3650 accepts your typed word which is then compared with the requested word. If the two words match, the ending time is read from the real-time clock, the starting time subtracted from it, and the resulting elapsed time is used to calculate your ranking as a marksman.

If the word you entered did not match the requested word (the comparison is done in uppercase), the routine then checks to see whether you might have typed the word in lowercase letters. Lines 3660–3670 convert the letters of the requested word. If there is still no match, you are told, "That's not it. Try again."

At the beginning of the program when you were asked to rank your shooting ability (Lines 600–700), the number that you input (HX) effectively becomes the number of seconds you have to type the requested shooting word in order to be ranked as an excellent marksman. Hence, if you claimed to be able to "hit a charging boar at 300 paces," in order to hit a bandit or animal dead center, you will have to type the correct word in one second or less. Actually, because of the way the real-time clock works, the allowable time may be as much as 1.99 seconds. Of course, you can "cheat" and rank yourself as a poor marksman (which will give you more time to type the correct word), but doing so makes the game less challenging and fun.

Many of the shorter subroutines are explained in the program notes in the next three chapters. Be sure to read them if you are having trouble understanding how anything works in this program.

PROGRAM VARIABLES

A	Answer to input query, numeric
A\$	Answer to input query, string
A1, A2	Upper and lower limit to input answer
B	Beasts (camels)
BA	Beast quality
BL	Beast load capacity
BSK	Beast sickness indicator
C	Clothes (changes)
CZ	No clothes indicator
D	Distance (miles per trip segment)

DT	Distance, total
DZ	Desert indicator
EP(n)	Event probabilities (n = 1, 14)
F	Food (sacks)
FA\$(n)	Food, names of animals for hunting
FC	Food carrying capacity of beast
FE	Food eaten on current trip segment
FP	Food eaten on previous trip segment
FQ	Food quality on current and previous trip segment
FR	Food reserve
HX	Hunting expertise level
I	Iteration variable
J	Trip segments (2-month periods)
K	Iteration variable
JL	Jewels
L	Oil (skins)
M	Medicines (bottles of balm)
MO	Month
MO\$(n)	Month name (n = -1, 6)
PFD	Person food indicator
PSK	Person sickness indicator
PSKT	Person sickness total
PWD	Person wound indicator
PWDT	Person wounds total
R	Rate of speed
RN	Random number variable
S\$(n)	Shooting words (n = 1, 4)
S1, S2	Shooting timer start and stop
SR	Shooting response
W	Weapons (crossbow arrows)
X\$	Temporary string variable
XA\$	Temporary string variable
YR	Year

PROGRAM LISTINGS

[Download MARCO.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF
110 LOCATE 10, 1 : X$ = "The Journey of Marco Polo, 1271" : GOSUB 3760
120 LOCATE 13, 1 : X$ = "(c) David H. Ahl, 1986" : GOSUB 3760
130 LOCATE 23, 1 : GOSUB 3720 : CLS
140 DIM EP(20)
150 JL = 300 : C = 2 : W = 30 : M = 5 : FP = 5 : BSK = 99 : 'Initial quantities of stuff
160 GOSUB 360 : GOSUB 3560 : 'Display the scenario
170 WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN
180 PRINT : GOSUB 720 : 'Purchase initial supplies
190 GOSUB 600 : 'Input hunting skill level
195 X$ = "Press any key to begin your trek!" : GOSUB 3760 : GOSUB 3740 : PRINT
200 '
210 'Main program
220 J = J + 1 : GOSUB 3510 : 'Next two-month segment
230 DT = DT + D : IF DT > 6000 THEN 3360 : 'Reached end of trip?
240 D = 40 + BA * 20 + INT(100 * RND(1)) : PRINT "You have traveled" DT "miles."
245 PRINT "Here is what you now have :" : GOSUB 3200
250 GOSUB 910 : 'Check for no jewels or clothes
260 GOSUB 1020 : 'Check for sickness
270 IF BSK = J THEN BSK = 99 : BL = B : BA = BA + 1 : 'Camel recover yet?
280 IF J > 1 AND JL > 1 THEN GOSUB 1190 : 'Barter for supplies
290 IF C = 0 THEN GOSUB 1400 : 'No clothes?
300 GOSUB 1500 : 'Eating routine
310 IF DZ = 0 AND RND(1) < .18 THEN GOSUB 3020 : '18% chance to hunt for food
320 PRINT : GOSUB 1780 : 'Desert sections
330 IF DZ = 0 THEN GOSUB 1940 : 'Event happens
340 GOSUB 3110 : GOTO 220
350 '
360 'Subroutine to print initial scenario
370 X$ = "The Journey of Marco Polo-1271" : GOSUB 3760 : PRINT : PRINT
380 PRINT " Starting from Venice in 1271 you travel by sailing ship to the"
390 PRINT "port of Armenia. Upon arrival, you prepare for a 6000-mile trek to"
400 PRINT "the court of the Great Kublai Khan in Shang-tu, Cathay. Having set"

```



```

410 PRINT "aside" JL "precious jewels to finance your planned 3-year trip, you"
420 PRINT "must barter for the following supplies in Armenia :"
450 PRINT " * Camels (Sturdier animals will cost more. You will probably"
460 PRINT " want 8 to 10 camels to carry your many supplies.)"
470 PRINT " * Food (You must barter for food as you travel along. However,"
480 PRINT " prices tend to be lower in port cities, so you should pack"
490 PRINT " in a good supply at the start.)
500 PRINT " * Oil for lamps and cooking (Over much of the trip, you will be"
510 PRINT " able to use wood to build fires. However, in the Persian,"
520 PRINT " Lop, and Gobi deserts you will need oil.)" : PRINT
522 PRINT " From Venice you have also packed clothing, weapons (crossbows),"
524 PRINT "and medicines (balms and unguents); however, your provisions will be"

530 PRINT "depleted as you go along and you must replenish them. The selection"
540 PRINT "and price of supplies is quite different in various regions, so you"
550 PRINT "must barter wisely. As a merchant, you are not skilled in fishing"
560 PRINT "or hunting, although occasionally you might be able to try to get"
570 PRINT "some food in this way."
580 GOSUB 3720 : PRINT : RETURN
590 '
600 'Subroutine to initialize hunting skill level
610 S$(1) = "SPLAT" : S$(2) = "SPRONG" : S$(3) = "TWACK" : S$(4) = "ZUNK"
620 FA$(1) = "wild boar" : FA$(2) = "big stag" : FA$(3) = "black bear"
625 PRINT : PRINT "Before you begin your journey, please rank your skill with"
630 PRINT "the crossbow on the following scale:"
640 PRINT " (1) Can hit a charging boar at 300 paces"
650 PRINT " (2) Can hit a deer at 50 paces"
660 PRINT " (3) Can hit a sleeping woodchuck at 5 paces"
670 PRINT " (4) Occasionally hit own foot when loading
680 INPUT "How do you rank yourself";HX
690 IF HX > 1 AND HX < 5 THEN PRINT : RETURN
700 PRINT "Please enter 1, 2, 3, or 4" : GOTO 680
710 '
720 'Subroutine to get initial supplies
730 PRINT " After three months at sea, you have arrived at the seaport of"
740 PRINT "Laiassus, Armenia. There are many merchants in the port city and"
750 PRINT "you can easily get the supplies you need. Several traders offer you"
760 A1 = 17 : A2 = 24 : PRINT "camels at prices between" A1 "and" A2 "jewels each."
770 INPUT "How much do you want to pay for a camel";A : GOSUB 3790 : BA = A
780 PRINT "You will need at least 7 camels, but not more than 12."
790 A1 = 7 : A2 = 12 : INPUT "How many camels do you want to buy";A : GOSUB 3790
800 B = A : JL = JL - BA * B : A2 = 3 * B - 6 : 'Camels-number, cost, amount they can carry
810 PRINT " One large sack of food costs 2 jewels. You will need at least"
820 PRINT "8 sacks to get to Babylon (Baghdad); you can carry a maximum of" A2
830 A1 = 8 : INPUT "sacks. How many do you want";A : GOSUB 3790
840 F = A : JL = JL - A * 2 : A2 = 3 * B - A : 'Food & cost, amount of oil camels can carry
850 PRINT " A skin of oil costs 2 jewels each. You should have at least 6"
860 PRINT "full skins for cooking in the desert. Your camels can carry" A2
870 A1 = 5 : INPUT "skins. How many do you want";A : GOSUB 3790
880 BL = B : L = A : JL = JL - 2 * L : 'Oil-amount and cost : return
900 '
910 'Subroutine to check for being out of jewels and clothes
920 IF JL > 15 THEN 980 : 'Still have a few jewels?
930 PRINT "You have only" JL "jewels with which to barter." : IF B > 2 THEN 950
940 PRINT "You push on with your" B "camels." : RETURN
950 INPUT "Would you like to sell a camel";A$ : GOSUB 3840 : IF A$ = "N" THEN 940
960 RN = INT(8 + 9 * RND(1)) : PRINT "You get" RN "jewels for your best camel."
970 JL = JL + RN : B = B - 1 : BL = BL - 1 : 'Add jewels, subtract camel

980 IF C > 0 THEN RETURN : 'Have some clothes?
990 PRINT "You should try to replace that tent you have been wearing as a"
1000 PRINT "robe. It is badly torn and the Tartars find it insulting." : RETURN
1010 '
1020 'Subroutine to deal with sickness
1030 IF PSK > 0 THEN PSKT = PSKT + PSK : PSK = 0 : 'Sickness total
1040 IF PWD > 0 THEN PWDT = PWDT + PWD : PWD = 0 : 'Injuries total
1050 IF FE = 3 THEN PFD = PFD + .4
1060 IF PSKT + PWDT + PFD < 3 THEN RETURN
1070 IF RND(1) > .7 THEN RETURN : '70% chance of delay due to recurring illness
1080 PRINT "As a result of sickness, injuries, and poor eating, you must stop"
1090 PRINT "and regain your health. You trade a few jewels to stay in a hut."

```

```

1100 RN = INT(1 + 3.2 * RND(1)) : IF RN > 3 THEN 1160 : '6% chance of dying
1110 PRINT "You grow steadily stronger, but it is" RN * 2 "months until you"
1120 PRINT "are again fit to travel." : PSKT = 0 : PWDT = 0 : PFD = 0 : J = J + RN
1130 M = INT(M / 2) : F = F / 2 : IF F < 3 THEN F = 3
1140 IF JL > 20 THEN JL = JL - 10 ELSE JL = INT(JL / 2) : 'Costs money for lodging
1150 GOSUB 3510 : RETURN
1160 FOR I = 1 TO 2500 : NEXT : PRINT "You stay for" RN "months but grow"
1170 PRINT "steadily weaker and finally pass away." : J = J + RN : GOTO 3320
1180 '
1190 'Subroutine to barter for supplies
1200 PRINT "You have" JL; : INPUT "jewels. Do you want to barter here";A$
1210 GOSUB 3840 : IF A$ = "N" THEN 1380
1220 RN = INT(17 + 8 * RND(1)) : PRINT "Camels cost" RN "jewels here. ";
1230 A1 = 0 : A2 = INT(JL / RN) : INPUT "How many do you want";A : GOSUB 3790
1240 B = B + A : BL = BL + A : BA = BA - A : 'Lower quality animals along route
1250 JL = JL - A * RN : RN = INT(2 + 4 * RND(1)) : PRINT "Sacks of food cost" RN "jewels. ";
1260 A2 = (INT(JL / RN)) : INPUT "How many do you want";A : GOSUB 3790 : F = F + A
1270 IF F + L > 3 * BL THEN PRINT "Camels can't carry that much." : F = F - A : GOTO 1260
1280 JL = JL - A * RN : RN = INT(2 + 4 * RND(1)) : PRINT "Skins of oil cost" RN "jewels. ";
1290 A2 = (INT(JL / RN)) : INPUT "How many do you want";A : GOSUB 3790 : L = L + A
1300 IF F + L > 3 * BL THEN PRINT "Camels can't carry that much." : L = L - A : GOTO 1290
1310 JL = JL - A * RN : RN = INT(8 + 8 * RND(1)) : PRINT "A set of clothes costs" RN;
1320 A2 = (INT(JL / RN)) : INPUT "jewels. How many do you want";A : GOSUB 3790
1330 C = C + A : JL = JL - A * RN : PRINT "You can get a bottle of balm for 2 jewels. ";
1340 A2 = JL / 2 : INPUT "How many do you want";A : GOSUB 3790 : JL = JL - 2 * A : M = M + A
1350 A2 = JL : RN = INT(6 + 6 * RND(1)) : PRINT "You can get" RN "arrows for 1 jewel."
1360 INPUT "How many jewels do you want to spend on arrows";A : GOSUB 3790
1370 JL = JL - A : W = W + RN * A : IF C > 1 THEN CZ = 0
1380 PRINT : PRINT "Here is what you now have:" : GOSUB 3200 : RETURN
1390 '
1400 'Subroutine to deal with no clothes
1410 PRINT : PRINT "You were warned about getting more modest clothes."
1420 PRINT "Furthermore, your sandals are in shreds." : IF CZ = 1 THEN 1470

1430 PRINT "The Tartars chase you from town and ";
1440 IF RND(1) > .2 THEN PRINT "warn you not to return." : CZ = 1 : RETURN
1450 PRINT "stone you." : PRINT "You are badly wounded and vow to get";
1460 PRINT "new clothes as soon as possible." : PWD = 1.5 : CZ = 1 : RETURN
1470 PRINT "Word has been received about your disreputable appearance."
1480 PRINT "The people are not willing to deal with you and they "; : GOTO 1450
1490 '
1500 'Subroutine to eat
1510 IF F < 3 THEN GOSUB 1650 : 'Out of food?
1520 PRINT "On the next stage of your journey, how do you want to eat : "
1530 PRINT " (1) Reasonably well (can walk further; Less chance of sickness)"
1540 INPUT " (2) Adequately, or (3) Poorly";A : IF A > 0 AND A < 4 THEN 1560
1550 PRINT "That's not a choice. Now then, (1) Well,"; : GOTO 1540
1560 FE = 6 - A : IF FE <= F THEN 1580
1570 PRINT "You don't have enough food to eat that well. Try again." : GOTO 1520
1580 FR = INT(.5 + 10 * (F - FE)) / 10 : IF FR > 3 THEN 1630
1590 IF FR = 1 THEN X$ = "" ELSE X$ = "s"
1600 PRINT "Your food reserve will then be just" FR "sack" X$ : IF A = 3 THEN 1630
1610 INPUT "Do you want to change your mind about how much you will eat";A$
1620 GOSUB 3840 : IF A$ = "Y" THEN 1520
1630 F = F - FE : D = D - (A - 1) * 50 : FQ = FP + FE : FP = FE : RETURN
1640 '
1650 'Out of food section
1660 PRINT "You don't have enough food to go on."
1670 IF JL < 15 THEN 1730
1680 PRINT "You should have bought food at the market. Now it will cost you"
1690 RN = INT(5 + 4 * RND(1)) : PRINT RN "jewels per sack."; : A1 = 1 : A2 = (INT(JL / RN))
1700 INPUT " How many sacks do you want";A : GOSUB 3790
1710 F = F + A : JL = JL - A * RN : IF F >= 3 THEN RETURN
1720 PRINT "You still don't have enough food and there is nothing to hunt."
1730 IF B < 1 THEN 1760 ELSE INPUT "Do you want to eat a camel";A$
1740 GOSUB 3840 : IF A$ = "N" THEN 3280 ELSE B = B - 1 : RN = INT(3 + 2 * RND(1)) : F = F + RN
1750 PRINT "You manage to get about" RN "sacks of food out of it." : RETURN
1760 PRINT "You don't even have a camel left to eat." : GOTO 3280
1770 '
1780 'Subroutine for desert sections

```

```

1790 DZ = 0 : IF DT < 2100 OR DT > 5900 THEN RETURN : 'No desert at far ends
1800 IF DT > 2600 AND DT < 4100 THEN RETURN : 'Tigris River Valley
1810 IF DT > 4600 AND DT < 5400 THEN RETURN : 'No desert in middle
1820 IF DT < 4100 THEN X$ = "Dasht-e-Kavir (Persian)" : GOTO 1840
1830 IF DT > 5399 THEN X$ = "Gobi (Cathay)" ELSE X$ = "Taklimakan (Lop)"
1840 PRINT "You are in the " X$ " desert."
1850 IF L >= 3 THEN L = L - 3 : PRINT "Use 3 skins of oil for cooking." : GOTO 1900
1860 PRINT "You ran out of oil for cooking."
1870 IF L > 1 THEN IF RND(1) > .5 THEN L = 0 : GOTO 1900

1880 PRINT "You get horribly sick from eating raw and undercooked food."
1890 L = 0 : PSK = 1 : D = D - 80 : M = M - 1
1900 ON INT(1 + 7 * RND(1)) GOSUB 2250, 2310, 2420, 2450, 2480, 2510, 1920
1910 DZ = 1 : GOSUB 3110 : RETURN
1920 PRINT "You get through this stretch of desert without mishap!" : GOTO 1910
1930 '
1940 'Subroutine to deal with special events
1950 RN = INT(EPT * RND(1)) : FOR I = 1 TO 14 : 'Iterate thru possible events
1960 IF RN > EP(I) THEN NEXT I : I = 14 : 'If event happened, exit lop
1970 IF I > 10 THEN 1990
1980 ON I GOTO 2000, 2250, 2310, 2340, 2360, 2380, 2400, 2420, 2450, 2480
1990 ON I - 10 GOTO 2540, 2570, 2600, 2660
2000 PRINT "A camel injures its leg. Do you want to (1) Nurse it along or"
2010 INPUT "(2) Abandon it, or (3) Sell it";A
2020 IF A = 1 THEN 2040 ELSE IF A = 2 THEN 2050 ELSE IF A = 3 THEN 2090
2030 PRINT "That is not a choice. Answer (1) to Nurse it along, " : GOTO 2010
2040 BSK = J + 2 : GOSUB 2120 : RETURN
2050 B = B - 1 : GOSUB 2120 : FC = 3 * BL - F - L : IF FC <= 0 THEN RETURN
2060 PRINT "You kill the camel for food." : IF FC > 2 THEN FC = 3
2070 F = F + FC : IF FC = 1 THEN X$ = "" ELSE X$ = "s"
2080 PRINT "You get the equivalent of" FC "sack" X$ " of food." : RETURN
2090 B = B - 1 : PRINT "It is a poor beast and you can get only 10 jewels for it."
2100 JL = JL + 10 : GOSUB 2120 : RETURN
2110 '
2120 'Exceed load carrying capacity of camels?
2130 BL = B : IF BSK <= J THEN BL = B - .6 : BA = BA - 1 : 'If sick camel reduce load, speed
2140 IF F + L <= 3 * BL THEN RETURN
2150 PRINT "You have too large a load for your camels." : FC = INT(F + L - 3 * BL + .9)
2160 IF FC = 1 THEN X$ = "" ELSE X$ = "s"
2170 PRINT "You'll have to sell" FC "sack" X$ " of food or skin" X$ " of oil."
2180 FS = INT(FC / 2) : LS = FC - FS : 'How much to sell of food and oil
2190 IF LS > L THEN LS = LS - 1 : FS = FS + 1 : GOTO 2190
2200 IF FS > F THEN FS = FS - 1 : LS = LS + 1 : GOTO 2200
2210 F = F - FS : L = L - LS : JL = JL + FS + LS : 'Decrease food and oil, add jewels
2220 PRINT "You sell" FS "of food," LS "of oil for which you get only";
2230 PRINT FS + LS "jewel" X$ "." : RETURN
2240 '
2250 PRINT "One of your camels is very sick and can't carry a full load."
2260 INPUT "Want to (1) Keep it with you, (2) Slaughter it, or (3) Sell it";A
2270 IF A = 1 THEN 2290 ELSE IF A = 2 THEN 2050 ELSE IF A = 3 THEN 2090
2280 PRINT "That is not a choice. Again, please." : GOTO 2260
2290 BSK = J + 2 : GOSUB 2120 : RETURN
2300 '
2310 PRINT "Long stretch with bad water. Costs time to find clean wells."
2320 D = D - 50 : RETURN

2330 '
2340 PRINT "You get lost trying to find an easier route." : D = D - 100 : RETURN
2350 '
2360 PRINT "Heavy rains completely wash away the route." : D = D - 90 : RETURN
2370 '
2380 PRINT "Some of your food rots in the humid weather." : F = F - 1 : RETURN
2390 '
2400 PRINT "Marauding animals got into your food supply." : F = F - 1 : RETURN
2410 '
2420 PRINT "A fire flares up and destroys some of your food and clothes."
2430 F = F - .4 : C = C - 1 : GOSUB 3110 : IF L < 1 THEN RETURN ELSE L = L - .5 : RETURN
2440 '
2450 PRINT "Two camels wander off. You finally find them after spending"
2455 PRINT "several days searching for them."
2460 D = D - 20 : RETURN

```

```

2470 '
2480 PRINT "You get a nasty burn from an oil fire."
2490 PWD = .5 : GOSUB 2840 : RETURN
2500 '
2510 PRINT "High winds, sand storms, and ferocious heat slow you down."
2520 D = D - 70 : RETURN
2530 '
2540 PRINT "A gash in your leg looks infected. It hurts like the blazes."
2550 GOSUB 2840 : D = D - 50 : PWD = .7 : RETURN
2560 '
2570 PRINT "Jagged rocks tear your sandals and clothing. You'll have to get"
2580 PRINT "replacements as soon as you can." : C = C - 1 : D = D - 30 : RETURN
2590 '
2600 RN = RND(1) * FQ : IF RN < 2 THEN 2610 ELSE IF RN < 3.5 THEN 2630 ELSE RETURN
2610 PRINT "All of you have horrible stomach cramps and intestinal disorders"
2620 PRINT "and are laid up for over a month." : D = D - 275 : RETURN
2630 PRINT "You're running a high fever and your muscles feel like jelly."
2640 PRINT "Your party slows down for you." : PSK = .7 : D = D - 125 : RETURN
2650 '
2660 PRINT "Blood-thirsty bandits are attacking your small caravan!"
2670 PRINT "You grab your crossbow..."; : GOSUB 3620
2680 IF W > 5 THEN 2700 ELSE PRINT "You try to drive them off, but you run out"
2690 PRINT "of arrows. They grab some jewels and food." : F = F - 1 : GOTO 2720
2700 IF SR <= 1 THEN 2810 ELSE IF SR <= 3 THEN 2780
2710 PRINT "Better stick to trading; your aim is terrible."
2720 IF RND(1) > .8 THEN 2750 : '80% chance of surviving attack
2730 PRINT "They are savage, evil barbarians - they kill you and take"
2740 PRINT "your remaining camels and jewels." : JL = 0 : B = 0 : GOTO 3320
2750 PRINT "You caught a knife in the shoulder. That's going to take quite"
2760 PRINT "a while to heal." : GOSUB 2840

2770 PWD = 1.5 : JL = JL - 10 : W = W - 4 - 2 * SR : GOSUB 3110 : RETURN
2780 PRINT "With practice you could shoot the crossbow, but most of your shots"
2790 PRINT "missed. An iron mace got you in the chest. They took some jewels."
2800 PWD = 1 : JL = JL - 5 : GOSUB 2840 : W = W - 3 - 2 * SR : GOSUB 3110 : RETURN
2810 PRINT "Wow! Sensational shooting. You drove them off with no losses."
2820 W = W - 4 : RETURN
2830 '
2840 'Subroutine to deal with using balm
2850 RN = INT(1 + 2 * RND(1)) : IF RN > 1 THEN X$ = "s" ELSE X$ = ""
2860 IF RND(1) > .5 THEN XA$ = "balm" ELSE XA$ = "unguent"
2870 M = M - RN : IF M < 0 THEN M = 0 : GOTO 2890
2880 PRINT "You use" RN "bottle" X$ "of" XA$ "treating your wound." : RETURN
2890 PRINT "You need more " XA$ "to treat your wound." : IF JL < 8 THEN 2940
2900 PRINT "Fortunately, you find some nomads who offer to sell you 2 bottles"
2910 PRINT "of" XA$ "for the outrageous price of 4 jewels each."
2920 INPUT "Do you want to buy it"; A$ : GOSUB 3840 : IF A$ = "N" THEN 2950
2930 PRINT "It works well and you're soon feeling better." : M = 0 : JL = JL - 8 : RETURN
2940 PRINT "But, alas, you don't have enough jewels to buy any."
2950 PRINT "Your wound is badly infected, "; : IF RND(1) < .8 THEN 3000
2960 PRINT "but you keep going anyway." : PRINT
2970 PRINT "Unfortunately, the strain is too much for you and, after weeks of"
2980 PRINT "agony, you succumb to your wounds and die in the wilderness."
2990 GOTO 3320
3000 PRINT "but you push on for the next village." : PWD = 3 : RETURN
3010 '
3020 'Subroutine to hunt for food
3030 IF W < 15 THEN PRINT "You don't have enough arrows to hunt for food." : RETURN
3040 PRINT "There goes a " FA$(INT(1 + 3 * RND(1))) "..."; : W = W - 15 : GOSUB 3620
3050 IF SR <= 1 THEN 3080 ELSE IF SR <= 3 THEN 3070
3060 PRINT "Were you too excited? All your shots went wild." : RETURN
3070 PRINT "Not bad; you finally brought one down." : FA = 2 : GOTO 3090
3080 PRINT "With shooting that good, the Khan will want you in his army." : FA = 3
3090 PRINT "Your hunting yields" FA "sacks of food." : F = F + FA : RETURN
3100 '
3110 'Subroutine to check for zero quantities
3120 IF JL < 0 THEN JL = 0 : 'Can't have negative jewels
3130 IF F < 0 THEN F = 0 : 'or food
3140 IF L < 0 THEN L = 0 : 'or oil
3150 IF C < 0 THEN C = 0 : 'or clothing
3160 IF M < 0 THEN M = 0 : 'or medicine

```

```

3170 IF W < 0 THEN W = 0 : 'or arrows
3180 RETURN
3190 '
3200 'Subroutine to print inventory
3210 PRINT TAB(22) "Sacks of Skins of Robes and Balms and Crossbow"

3220 PRINT "Jewels Camels Food Oil Sandals";
3230 PRINT "Unguents Arrows" : GOSUB 3110
3240 PRINT USING "#####";JL; : X$ = "#####": XAS = "#####.#"
3250 PRINT USING X$; B; : PRINT USING XAS; F; : PRINT USING XAS; L;
3260 PRINT USING X$; C; : PRINT USING X$; M; : PRINT USING X$; W : PRINT : RETURN
3270 '
3280 'End game - out of food
3290 PRINT "You keep going as long as you can, trying to find berries and"
3300 PRINT "edible plants. But this is barren country and you fall ill and,"
3310 PRINT "after weeks of suffering, you collapse into eternal sleep."
3320 PRINT : J = J + 1 : GOSUB 3510 : PRINT "You had the following left at the end :"
3330 GOSUB 3200 : PRINT "You traveled for" J * 2 "months!"
3340 PRINT : PRINT "Sorry, you didn't make it to Shang-tu." : GOTO 3490
3350 '
3360 'End of trip section
3370 GOSUB 3110 : 'Can't have negative jewels at end
3380 FOR I = 1 TO 3000 : NEXT I : CLS : FOR I = 1 TO 10
3390 BEEP : X$ = "CONGRATULATIONS !" : LOCATE 12, 1 : GOSUB 3760
3400 FOR K = 1 TO 100 : NEXT K : CLS : FOR K = 1 TO 50 : NEXT K : NEXT I
3410 CLS : PRINT "You have been traveling for" J * 2 "months !" : PRINT
3420 PRINT "You are ushered into the court of the Great Kublai Khan."
3430 PRINT "He surveys your meager remaining supplies :" : GOSUB 3200
3440 PRINT "... and marvels that you got here at all. He is disappointed"
3450 PRINT "that the Pope did not see fit to send the 100 men of learning"
3460 PRINT "that he requested and, as a result, keeps the three of you as"
3470 PRINT "his personal envoys for the next 21 years. Well done!" : PRINT
3480 '
3490 PRINT : INPUT "Would you like to try again"A$ : GOSUB 3840
3500 IF A$ = "Y" THEN RUN ELSE CLS : KEY ON : PRINT "Bye for now." : END
3505 '
3510 'Subroutine to print the date
3520 MO = J : WHILE MO > 6 : MO = MO - 6 : WEND
3530 YR = 1271 + INT(J / 6)
3540 PRINT : PRINT "Date :" MO$(MO) YR : RETURN
3550 '
3560 'Subroutine to read event probabilities
3570 FOR I = 1 TO 14 : READ A : EPT = EPT + A : EP(I) = EPT : NEXT I
3580 DATA 6, 4, 4, 6, 6, 6, 6, 4, 1, 6, 8, 18, 10
3590 FOR I = 1 TO 6 : READ MO$(I) : NEXT I : RETURN
3600 DATA "March", "May", "July", "September", "November", "January"
3610 '
3620 'Subroutine to shoot crossbow
3630 RN = 1 + INT(4 * RND(1)) : 'Print random shooting word
3640 S1 = 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) : 'Start timer
3650 PRINT "Type :" S$(RN) " " : INPUT X$ : IF X$ = S$(RN) THEN 3680

3660 FOR I = 1 TO LEN(X$) : 'Iterate through letters for possible lowercase
3670 IF MID$(S$(RN), I, 1) < > CHR$(ASC(MID$(X$, I, 1)) - 32) THEN 3700 ELSE NEXT I
3680 S2 = 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) : 'End timer
3690 SR = S2 - S1 - HX : RETURN : 'Shooting response
3700 PRINT "That's not it. Try again." : GOTO 3650
3710 '
3720 'Subroutine to hit continue key
3730 X$ = "Press any key to continue." : GOSUB 3760
3740 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND : RETURN
3750 '
3760 'Subroutine to print a centered line
3770 PRINT TAB((70 - LEN(X$)) / 2) X$ : RETURN
3780 '
3790 'Subroutine to check if answer entered is in range
3800 IF A >= A1 AND A <= A2 THEN RETURN
3810 IF A < A1 THEN X$ = "few" ELSE X$ = "many"
3820 PRINT "That is too" X$ : INPUT ". Your answer please"; A : GOTO 3800
3830 '
3840 'Subroutine to process a yes/no answer

```

```
3850 GOSUB 3880 : IF A$ = "Y" OR A$ = "N" THEN RETURN
3860 INPUT "Don't understand answer. Enter 'Y' or 'N' please";A$ : GOTO 3850
3870 '
3880 'Subroutine to extract the first letter of an answer
3890 IF A$ = "" THEN A$ = "Y" : RETURN
3900 A$ = LEFT$(A$, 1) : IF A$ >= "A" AND A$ <= "Z" THEN RETURN
3910 A$ = CHR$(ASC(A$) - 32) : RETURN
```

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



Discover more

[Hiking & Camping](#)

[Computer Hardware](#)

[Computer](#)

[Rail Freight](#)

[Data Management](#)

[Hunting & Shooting](#)

[clothes](#)



THE GAME

In *Westward Ho!* you are the head of a family of five setting out from Independence, Missouri, in the spring of 1847 on the Oregon Trail. Your objective is to arrive safely in Oregon City, Oregon.

Having saved about \$420 you must purchase a wagon for \$70, and with the remaining money you must also buy enough equipment, supplies, and livestock to sustain you on your strenuous 2000-mile journey. As you travel, you encounter the same hazards and conditions that American pioneers faced during the Great Migration: wagon fires, polluted water, wild animals, bad weather, illness, and topographical obstacles. The following hints will help you on the Trail:

- In Independence you decide how much of your money to spend on the things you need for the trip. You can spend all of your money there, but if you do, you will not be able to buy supplies at forts along the way.
- The trip is divided into two-week segments. Between segments you have an opportunity to stop and hunt for game. Hunting may augment your food supply, but it always consumes valuable time.
- At the beginning of the game, you are asked to rank your shooting (typing) ability. When you are hunting or being attacked by hostile Indians, you are asked to type a word that sounds like a gunshot; the faster you type it, the more likely you are to hit your target. Of course, not all Indians that approach your wagon are hostile, and shooting at friendly Indians costs you time and ammunition.

If you make the correct decisions along the way, you and your family will join the hundreds of thousands of pioneers who settled and developed the land west of the Rockies. If you prepare poorly or if you make foolish decisions, your bones will serve as a warning to those who come after you on the Oregon Trail. Good luck!

Monday, April 26, 1847. You are near Fort Kearney.
Total mileage to
Here's what you

THE STORY

No one can say when it began. A thousand little rivulets of water trickling downhill go unnoticed until they merge into a thundering river. So it was at the beginning of the greatest mass migration of people that this country, or any other, has ever known. There was only one way to go: west. West to farmlands in Ohio, west along the Erie Canal to the Great Lakes, west to the tributaries of the Mississippi and Ohio rivers, and west to the rolling hills of Kentucky.

By 1825, steamboats were plying the Mississippi and Ohio rivers, and the states of Michigan, Illinois, and Indiana were considered settled. Settlers pushed west into Iowa and Missouri until, by 1830, Independence, Missouri, then the westernmost post office in the United States, defined the boundary of the frontier.

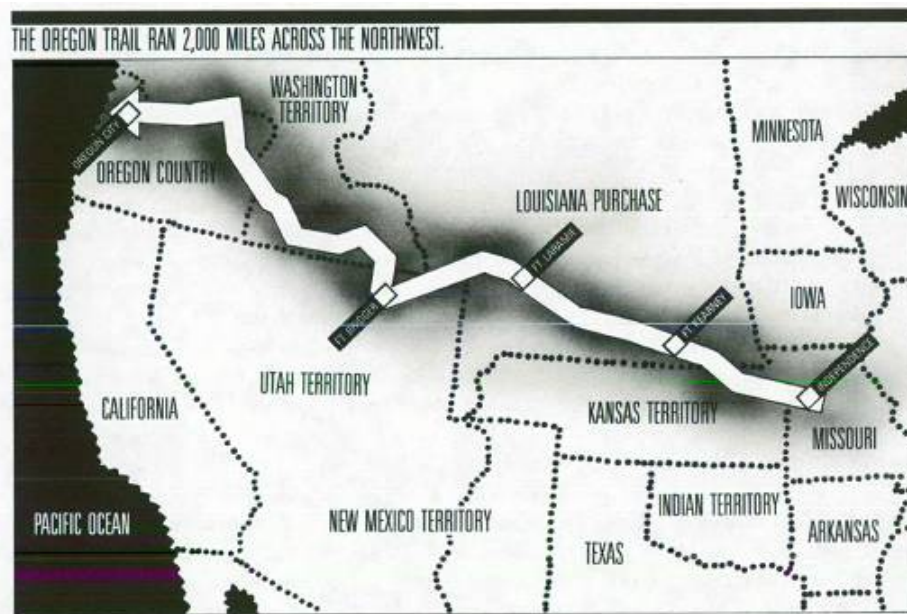
Independence was on the very threshold of the frontier, and for more than a decade it remained the focal point for the western march of the pioneers. Independence was ideally situated for such a role. It was three miles south of the big bend in the Missouri River, where, after flowing southeast for 2000 miles from its headwaters in the Rocky Mountains, the great river changed direction and flowed due east to join the Mississippi a few miles north of St. Louis. A few miles to the west began the vast undulating prairies and high plains that stretched unbroken to the distant Rockies.

With the lands east of the Mississippi settled, adventurers, either by choice or compulsion, gravitated to an even newer frontier—the lands west of the Mississippi. Like their fathers, they were traders, trappers, hunters, and explorers—almost certainly not farmers or settlers.

An early lure of the west was the lower Rio Grande, where inexpensive Spanish and Mexican linens and fabrics could be bought from Spanish traders. New England textile merchants smacked their lips at the thought of profits from capturing that trade. Trade with various Indian tribes was also of interest, and by 1825 a regular route across the Kansas prairie, along the Arkansas and Purgatoire Rivers, and across the Sangre de Cristo Mountains, had been established from Independence to Santa Fe, New Mexico. This, the first of the major trails west, was known as the Santa Fe Trail.

By 1830, adventurers had started looking westward to destinations other than New Mexico. In 1831, Joseph Smith, Jr., set out from Independence and, upon reaching what is now Salt Lake City, declared, "This is the place." Tens of thousands of Mormons followed in his footsteps over the next three decades to escape religious persecution in Ohio and Missouri.

Meanwhile, Stephen Watts Kearney, a determined U.S. Army general, had pushed west over a southern route from Santa Fe through Arizona to San Diego, and other people had extended the northern trails west across the mountains to San Francisco (the Overland Trail) and to Willamette Valley (the Oregon Trail). In addition, steamboats of the American Fur Company had been able to navigate up the Missouri as far as Fort Pierre, South Dakota, while smaller vessels had gotten to Fort Benton, Montana. To cement its foothold, this privately owned company was busy establishing outposts and forts throughout the northern states.



Independence, absorbed with the Santa Fe trade to the south and the fur trade to the north, had paid little attention to the trickle of emigrants who, for several years, had been setting out for California and Oregon. However, as glowing reports began to come back from the early pioneers, the trickle swelled to a flood—900 emigrants left Independence in 1843, 3000 in 1845, and more than 5000 in 1847. These new pioneers bore little resemblance to the traders plying the Santa Fe Trail or to the trappers in the North. They were men of the land, traveling with their wives and children. They were movers, but they had a destination, a promised land called Oregon—about which they knew as little as they knew about the road that would take them there.

Historically, the Conestoga wagon, which originated in eastern Pennsylvania, has always been associated with the great migration to Oregon and is depicted in scores of paintings. The truth, however, is much less colorful. The Conestoga wagon was in fact far too heavy for the long haul across the prairies and mountains, and a flatbed farm wagon, sometimes fitted with high wheels and a tent of waterproof sheeting, was generally the transport of choice. Such a wagon, sometimes called a Murphy wagon, required a team of six to ten mules or oxen to pull it when heavily laden. Most families also had a cow or two, a saddle horse, and a plow lashed to the rear of the wagon.

A bare-bones Murphy wagon cost about \$50 to \$70; high wheels, a waterproof covering, yokes, harnesses, and spare parts could bring the total cost up to \$100. A team of two oxen cost about \$25; most travelers bought six to ten animals. Although horse teams could travel faster than oxen, ox teams were sturdy, dependable, and less likely to be stolen by Indians. And if worst came to worst and food ran out, the oxen could be eaten.

A guidebook of the time recommended the following food supplies for one adult for the five- to six-month journey: 150 lbs. of flour, 25 lbs. of bacon or pork, 15 lbs. of coffee, 25 lbs. of sugar, and smaller quantities of rice, beans, dried fruit, molasses, vinegar, salt, pepper, tea, spices, and baking soda. Also on the recommended list were tobacco, soap, whiskey, medicines, and matches.

It was essential that the wagon carry spare parts and tools such as oxen yokes, harnesses, lead bars, open chain links, horseshoes, nails, ropes, hammers, axes, mallets, saws, and spades. Tar buckets were also necessary, some filled with resin and grease to use on the axles, and others filled with tar to seal and waterproof the wagon before fording or floating it across a river.

When they arrived in Independence, most families already had some supplies and clothing. However, tales of harsh weather in the mountains inspired all but the most foolhardy travelers to procure additional warm clothing.

For protection as well as hunting, travelers carried breech loading rifles, Colt revolvers, and a plentiful supply of ammunition. Most also took along mirrors, ribbons, cloth, tobacco, and assorted trinkets for trading with the Indians.

Most families joined others in Independence and made up wagon trains of from four to as many as 100 wagons. Each season, the first of these trains set out as soon as the winter snows melted and was followed by a steady stream of departures throughout the spring and into early summer.

The wagon trains usually went only a short distance the first day, making a sort of trial run. While they were still close to Independence, the men could ride back if necessary to buy supplies that had been forgotten.

The first weeks of travel in the spring were generally very difficult. The men frequently were not experienced at handling teams, wagons, or weapons. The snows had just melted, so the trails were like mud bogs. Most families had no idea how to pack and wound up with dangerously top-heavy or unbalanced loads. Oxen became entangled in their ropes at night and sometimes, breaking free, wandered off.

The first stop for most travelers, usually reached in two or three days, was the Shawnee Methodist Mission only 15 miles from Independence. Once this was left behind, travelers would not see any signs of civilization for many long miles. After leaving the Mission, the wagon trains lumbered over the rolling prairie south of the Kansas River, following the deep ruts of the Santa Fe Trail as far as the present-day town of Gardner, Kansas. There a sign bore the simple legend "Road to Oregon."

From there on, the wagons kept to the high prairies as much as possible, although there were many streams and rivers to be crossed. The Wakarusa, Kansas, Red Vermillion, Black Vermillion, and Big Blue rivers were especially difficult for the travelers; smaller streams could be forded, but it was a daylong ordeal to cross a river. First, the wagons had to be unloaded and the joints and seams packed with tar. After that, they were let down the bank with ropes and floated across. Supplies were floated by makeshift raft or carried by horse. Frequently tools and heavier provisions slid into the river. After the oxen had crossed, they were taken to the top of the bank, harnessed by long ropes to the wagon, and, with them pulling and all the family members pushing, the wagon inched to the top of the bank.

Indians, particularly the Pawnee in the area of the Red Vermillion river, did not welcome the constant stream of white men crossing their hunting grounds. In 1849, for example, after a cholera epidemic for which the Indians blamed the whites, the Pawnee, Oglala, and Sioux began attacking wagon trains with great frequency. Wagons were particularly vulnerable when crossing rivers, so the Indians often chose fords for their attacks.

In a sense, the Indians were correct about the source of cholera. It had been carried from Asia to the U.S. by sailors and passengers on ships. It reached the frontier by way of New Orleans and the Mississippi and traveled west with the wagon trains. No amount of planning or preparation could save the settlers from this hazard. Afflicted with severe pain, vomiting, and cramps, a person might display the first symptoms in the morning and be dead by noon.

In 1852, Ezra Meeker kept a log and estimated that more than 5000 people had died of cholera on the trail that year.

Living in fear of disease, the emigrants were prone to dose themselves with large quantities of medicine at the first sign of any illness, on the theory that the larger the dose the quicker the recovery that might be expected. In fact, many patients were killed rather than cured by the injudicious use of medicine.

River crossings slowed the journey through Kansas and Nebraska, and most wagon trains took about three weeks to travel the 175 miles to the ford across the Big Blue River in southern Nebraska known as Independence Crossing. About six miles northwest of this crossing, trails from St. Joseph and Fort Leavenworth converged, thence following the Platte River to Fort Kearney. Having reached the Platte, the pioneers could follow its valley west, past Chimney Rock, Scott's Bluff, and to the last outposts of civilization, Fort Laramie and Fort Fetterman.

The Platte River marked the beginning of buffalo country, and few men missed the chance to enjoy some good hunting and to add to their food supply. Unfortunately, most of them had no idea how to preserve the meat. The animals were generally shot in midafternoon and left in the hot sun until sunset, when they were gutted. Perhaps one was roasted that night, but the rest were left unskinned and undivided to rot and provide a meal for the scavengers of the plain.

Along the North Platte River Valley, the wagons could generally make a speed of about two miles per hour, thus covering, in a good day, about 15 to 18 miles. If it had been possible to maintain this speed for the entire journey, the 2040 miles from Independence to Oregon City could have been covered in about 4 1/2 months. However, everything seemed to conspire to slow the trip: river crossings, Indian and bandit attacks, hunting, burying the dead, wagon breakdowns, muddy trails, oxen wandering off, and losing the trail.

Some wagon trains even rested on Sunday, observing it as a day of worship and, more often than not, repair.

Fort Laramie stood at the fork of the Laramie and North Platte rivers in eastern Wyoming. There the traveler had his first opportunity in many weeks to send letters home, buy provisions, and get information about the trail ahead. There, too, was a place to relax a bit from the constant caution so necessary on the march.

After Fort Laramie, the next objective was the Sweetwater River Valley in central Wyoming, the entrance to which was marked by Independence Rock, on which thousands of emigrants carved their names. "The Great Register of the Desert," Father Pierre Jean de Smet, a Jesuit missionary, called it. If things were going well, most travelers had reached the rock by July 4.

Looking west from Independence Rock, the emigrants could see, six miles in the distance, a V-shaped split in a rocky ridge known as Devil's Gate. The preferred route skirted Devil's Gate Canyon, but there was no question that it marked the beginning of a new and more difficult phase of the journey. Although there were many alternative routes to the Sweetwater Valley, all trails converged there for the long ascent to South Pass across the Continental Divide. Even in mid-July, the steep mountain walls often blocked the sun, and snow and ice frequently covered the ground. The ascent from the Sweetwater Valley was long and gradual, and the South Pass many miles in width. Hence the only real hindrances to good progress were the occasional snow and damage to wagon wheels from the rocky trail. The travelers, having left the boredom of the plains and knowing that they had reached the halfway point of the trip, were usually in good spirits as they entered South Pass.

Beyond the South Pass, the trail began a gradual but rocky descent of about 60 miles, across the Green River near the Wyoming-Idaho border. This crossing was an extremely dangerous one; the river was wide, deep, powerful, and ice cold. Those who successfully made this crossing were much relieved to follow the Bear River Valley for a way to Soda Springs, whence they headed northwest for 50 miles to Fort Hall, Idaho, on the Snake River. There they were forced to decide whether to continue on to Oregon or turn south to California.

Fort Hall was a welcome stop for the wagon trains. Originally built by Captain Nathaniel J. Wyeth of Boston in 1834, the fort offered needed supplies and protection from the hostile Blackfoot Indians. After leaving the fort, the trail crossed the Portneuf River and Bannock Creek and then passed the American Falls on the Snake River. This treacherous crossing led, a few miles on, to two enormous rocks known as Massacre Rocks because hostile Indians often used them as a place from which to ambush wagon trains.

The trail then followed the Snake River for some 300 miles as it traced its circuitous course across the barren, lava-covered Snake River Plains; it finally broke out of the canyon at the mouth of the Little Boise River near the Oregon border. As they crossed the Snake at Fort Boise, the travelers took their first steps in Oregon.

But one more barrier loomed between the settlers and the Promised Land: the terrible Blue Mountains. Several times during the crossing of these mountains, wagons had to be lowered with ropes from one part of the trail to another, and canyon walls too narrow for the wagons frequently had to be chipped away. Wagons were often abandoned, their occupants continuing the journey on foot. Others, searching for a better route, became hopelessly lost and eventually died of hunger and thirst.

John Kerns, in his diary, recorded this about the Blue Mountains: "...it was the roughest road we have encountered on the journey, being up and down sidling mountains, into the brush and across a creek every 200 or 300 yards, and over stony places enough to hide all despairing sinners."

Those pioneers who traversed the final obstacle, Deadman's Pass, saw the trail emerge from the mountains and wind down the bald face of Emigrant Hill from which they were treated to one of the most spectacular views in the world. Mt. Hood and Mt. Adams could be seen in the distance, while in the foreground lay the rolling hills and fertile valleys of the Columbia River Basin.

From Walla Walla, Washington, on, still 250 miles across Oregon to the coastal settlements, most travelers kept going, driven by sheer exhilaration and determination. By all accounts, it was one of the most difficult parts of the journey. As Medorem Crawford, an emigrant traveler, recorded, "From Walla Walla to Willamette Falls (Oregon City) occupied about 20 days, and, all things considered, was the hardest part of the entire journey—what with drifting sands, rocky cliffs, and rapid streams along the Columbia, and the gorges, torrents, and thickets of the

Cascade Mountains, it seems incredible how, with our worn out and emaciated animals, we ever reached our destination." But many did, and they were well rewarded for their perseverance.

J.M. Shively, writer of one of the guidebooks about the trail, closed with the thought, "Be of good cheer—you will find a country in Oregon that will fill your desires, and repay you for all your toil."

BIBLIOGRAPHY

Coons, Frederica B. *The Trail to Oregon*. Portland, OR: Binfords & Mort, 1954.

Ghent, William J. *The Road to Oregon*. New York: Longmans, Green & Co., 1929.

Meeker, Ezra. *Ox Team Days on the Oregon Trail*. New York: E. Meeker, 1907.

Morgan, Dale L. *Overland in 1846*. Georgetown, CA: Talisman Press, 1963.

Parkman, Francis. *The Oregon Trail*. Boston, MA: Little, Brown & Co., 1925.

Rawitsch, Dan. *MECC Oregon User Manual*. Lauderdale, MN: Minnesota Educational Computer Consortium, 1977.

THE PROGRAM

Westward Ho! is a substantial revision of the Oregon Trail program written by Dan Rawitsch and Bill Heinemann in 1972–73 on the Hewlett Packard 2000 timesharing system of the Minnesota Educational Computer Consortium. The original Oregon Trail program was subsequently converted to Microsoft BASIC and appeared in *Creative Computing* magazine in 1977. Play of the game is similar in this new version, but the program has been structured (to some extent), simplified, and shortened by more than 300 lines.

The program consists of a main section, initialization and closing sections with five related subroutines, eight major subroutines, and six minor subroutines.

The initialization section dimensions variables, puts data in them, displays the initial scenario, and asks you for your initial purchases. At the outset, you have \$420, the average amount of money that a family setting out on the Oregon Trail had in 1847. A wagon costs \$70, and you can spend the rest of your money on oxen, food, ammunition, clothing, and miscellaneous supplies such as medicine, bandages, and repair parts. You can spend all of your money at the outset, or you can save some to spend at forts along the way to replenish supplies when they run low. This is a tradeoff: Supplies cost 50% more at the forts. However, if you lose supplies while fording a river or in a bandit attack, you may not have enough to continue.

Food is important to maintain the strength and health of your family. A guidebook of the time recommends for each adult 150 lbs. of flour, 25 lbs. of sugar, 25 lbs. of bacon, 15 lbs. of coffee, and smaller quantities of other staples. The average family of five (your family in the simulation) eats about as much as four adults. The above commodities in 1850 cost about 10 cents a pound in Missouri; thus an adequate food stock for your family would cost about \$60 to \$100. Of course, along the way you will hunt for fresh meat and, hopefully, find some edible plants and berries.

Although you are wearing some clothes, most travelers had to purchase additional clothing for the cold weather they would encounter in the mountains. In 1850, \$15 would buy several warm outfits; thus an appropriate budget for clothes would be at least \$50 or so for your family of five.

You will want to have plenty of ammunition for your rifle to hunt, ward off attacks by wild animals, and defend yourself against bandits and hostile Indians. A belt of bullets costs \$1, and in general you can expect to use one or two belts per week on the trail.

A brief explanation of these purchases is displayed in the subroutine in Lines 490–670, and the amounts are accepted as input in the subroutine at Lines 690–900. Note the conditional (IF) statements in this subroutine that do not allow you to buy less than the minimum you need, or to spend more than you have.

The main program (Lines 250–470) iterates through the journey in two-week segments. If you have been on the trail for more than 20 weeks, the program branches to an end-game routine. Under normal circumstances, at the beginning of a two-week trip segment, the date will be printed, injuries and illnesses treated (assuming you have enough money to pay a doctor), the mileage updated, and your inventory of supplies printed.

The expected mileage over the next two weeks is then calculated in Line 390. In general, you will travel 200 miles plus some additional distance which depends upon the quality of your team of oxen. This mileage figure is an ideal, assuming nothing goes wrong. If you run into problems, mileage is subtracted from this ideal figure; the revised total is printed at the start of the next trip segment.

You are then asked whether you wish to hunt, continue on, or, if it is an even-numbered trip segment, stop at a fort. In 1847, forts and missions were spaced about 300 to 400 miles apart along the Oregon Trail. On average you will cover about 75 miles a week and thus you can expect to hit a fort about every four weeks (or every other trip segment). Note the function in Line 1010 that determines whether or not you are on an even-numbered trip segment; if the integer value of $J/2$ equals $J/2$, J is even; if not, J is odd.

If you stop at a fort (Lines 1100–1180), you have an opportunity to purchase supplies, but they cost 50% more than at the start. The running total of most items in your inventory is kept in dollars. Your supply of ammunition, however, is expressed as a number of bullets. This makes it easier to calculate ammunition consumption when you use your gun to hunt, fight, or scare off animals.

If you decide to hunt, the subroutine at Lines 1200–1290 is called. If you have fewer than 40 bullets, you do not have enough to hunt and you are given the option of stopping at a fort (if there is one on this trip segment) or continuing on. Hunting costs several days of travel (45 miles) and, of course, some ammunition.

Since there is no good way to determine how skilled a marksman you are, the program asks you to rank yourself at the start of the trip (Lines 920–980). From then on, each time you are confronted with a situation in which you must use your rifle, you will be asked to type in a word that sounds like a gunshot (pow, blam, or bang).

The faster you type the word and hit Return, the better luck you will have in hitting your target.

The shooting subroutine is found at Lines 3870–3930. When this subroutine is entered, a random shooting word is selected in Line 3880. Then a timer is started. The timer uses the real-time clock of the computer to get a starting time in seconds; this equals 60 times the current clock minutes plus the current clock seconds. You then type in the shooting word, which is compared to the requested word. If your typing was in error, you are asked to type the word again. This continues until the word is typed correctly, at which time the time in seconds is again calculated. The starting time is subtracted from the ending time, and the resulting elapsed time in seconds less your ranking as a marksman is put into variable BR. A good typist should be able to type a four-letter word in two seconds or less, while a less skilled, hunt-and-peck typist might take four or five seconds.

Control is then returned to the hunting subroutine, and ammunition consumption calculated as a function of the value of BR. The slower you shoot (the higher the value of BR), the greater the probability that you are unsuccessful in your hunting (see function in Line 1240).

The eating subroutine is called next (Lines 1310–1370). In it you are asked how well you want to eat, and your food consumption is calculated in Line 1350. If you do not have enough food to eat as well as you would like, you must choose to eat at a diminished level.

Next, a subroutine is called to determine whether or not you are attacked by bandits or Indians (Lines 1390–1780). The probability of attack is determined by the function in Line 1400. Toward the beginning of your journey, especially around 400 to 500 miles from Independence in what is now western Nebraska and Wyoming, you have the highest probability of encountering bandits and Indians. As you get into the mountains, these probabilities decrease drastically (see graph on page 36). The function within the IF ... THEN statement increases the probability from 20% at 0 miles to a maximum of 50% at 500 miles and then gradually decreases the rest of the way.

If riders approach, you may choose one of four strategies: run, attack, ignore them, or circle wagons. Each strategy has a different cost in miles and supplies and also depends upon whether the riders were hostile or friendly. If you choose to attack, the program again goes to the shooting subroutine. If you are slow on the draw, you may pick up a flesh wound, which sets the injury flag and requires treatment by a doctor the next time you stop.

The longest subroutine (Lines 1800–2620) deals with hazards and special events. A random number selected in Line 1810 determines which event occurs; the program then branches to the appropriate routine to handle that event. The probability of each event is determined by the difference between successive numbers in the event array (Line 3720). For example, if a random number between 0 and 6 is selected, event 1 occurs; between 6 and 11, event 2; between 11 and 13, event 3; and so on. Thus we see that there is the highest probability that event 19 (value between 69 and 95) will occur; this event has to do with illness from not eating well.

Most events are handled in a very straightforward manner: A message is printed, mileage subtracted, and supplies used. On the other hand, four of the events are more complicated: cold weather, bandit attack, wild-animal attack, and illness.

The cold-weather routine (Lines 2130–2160) checks to see if you have adequate clothing to keep warm. If not, the illness subroutine (Lines 2880–2970) is called. This routine is also called if you are not eating well enough (Lines 2560–2590).

In the illness routine, depending upon how well you have been eating, you may contract a mild, bad, or very serious illness. Mild and bad illness can be treated with your own medicine—if you have any left—whereas serious illness requires the services of a doctor (the illness flag, KS, is set) at the start of the next trip segment.

The bandits attack routine (Lines 2180–2270) is very similar to the attack by Indians routine in the previous section, except that bandits are always bad and you have no choice but to fight them.

The last major subroutine called by the main program deals with travel through the mountains (Lines 2640–2860). In the mountains, you are vulnerable to cave-ins along the trail, losing your way, and just plain slow going. In addition, you must traverse the South Pass and the Blue Mountains. The probability is set to 80% that you will get stuck in the South Pass when you first reach it, but a flag (KP) prevents you from being stuck there for the entire journey. Likewise, you have a 70% chance of getting stuck in the Blue Mountains, but you will eventually get through. Blizzards, on the other hand, can occur on every trip segment while you are in the mountains, and there is a good chance that this will happen.

If you do not have adequate food, clothing, ammunition, or medical supplies when you encounter the various hazards of the journey, chances are very high that you will die on the trail (Lines 2990–3170). If you die, a short message is displayed telling you what happened, how far you traveled, and your remaining inventory of supplies. You are then given a chance to play again (Lines 3300–3330).

It is not known today what percentage of the travelers who set out on the Oregon Trail actually reached their destination. Certainly it was fewer than 50%, and perhaps fewer than 20%. If you are among the lucky few, the program will give you a congratulatory message (Lines 3190–3280), tell you how long it took, and display your remaining supplies, if any.

It is not easy to reach Oregon. Your decisions must be well reasoned, and Lady Luck must be traveling with you. But if you don't make it, you, unlike actual families in 1847, at least, will get a chance to try again.

PROGRAM VARIABLES

A	Money spent on animals
AS	Money to question (Y or N), user input
B	Money spent on ammo, also number of bullets
BR	Response time for typing shooting word
C	Money spent for clothing
C1	Flag for insufficient clothes
D	Total days traveled

DD	Days of last month
DM	Total months traveled
DA\$(n)	Date, n = 1 - 20
DR	Shooting expertise level
E	Eating quality
EP(n)	Event probability, n = 1 - 20
EV	Event counter
F	Money spent on food
GH	Riders description (0 hostile, 1 friendly)
GT	Choice of tactics when riders approach
I	Temporary iteration variable
J	Trip segment counter
KB	Flag for blizzard
KF	Flag for stop at fort
KH	Flag for injury
KM	Flag for Blue Mountains
KP	Flag for South Pass
KQ	Flag for not enough ammo to hunt
KS	Flag for illness
M	Total trip mileage
MA	Mileage through previous turn
MP	Mileage flag for South Pass
P	Amount spent at fort
R	Money for medicine and repair parts
RN	Random number for choosing events
S\$(n)	Shooting words, n = 1 - 4
S1, S2	Response time temporary variables
T	Cash
X	Choice of action, temporary
X\$	Temporary string variable

PROGRAM LISTINGS

[Download WESTHO.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF
110 LOCATE 10, 31 : PRINT "Westward Ho! 1847" : PRINT : PRINT
120 PRINT TAB(28) "(c) David H. Ahl, 1986" : LOCATE 23, 21
130 '
140 'Initialization
150 DIM DA$(20), EP(20), MP(15), PL$(15)
160 GOSUB 3430 : GOSUB 3530 : GOSUB 3700 : 'Put data in variables
170 PRINT "Press any key when you're ready to go" : RN = -32768!
180 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND
190 WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN : CLS
200 GOSUB 490 : 'Display the scenario
210 GOSUB 690 : 'Make initial purchases
220 GOSUB 920 : 'How good a shot are you?
230 PRINT : PRINT " Your trip is about to begin..." : PRINT : GOSUB 3830
240 '
250 'Main program
260 IF M > 2039 THEN 3190 : 'Reached the final segment?
270 J = J + 1 : 'Iterate through 2-week trip segments
280 IF J > 20 THEN 3080 : 'On the trail too long?
290 PRINT : PRINT "Monday, " DA$(J) ", 1847. You are ";
300 FOR I = 1 TO 15 : IF M > MP(I) THEN NEXT I ELSE PRINT PL$(I)
310 IF F < 6 THEN PRINT "You're low on food. Better buy some or go hunting soon."
320 IF KS < > 1 AND KH < > 1 THEN 370 : 'Any sickness or injuries?
330 T = T - 10 : IF T < 0 THEN 3010 ELSE PRINT "Doctor charged $10 for his services";
340 PRINT "to treat your "; : IF KS = 1 THEN PRINT "illness." ELSE PRINT "injuries."
350 KS = 0 : KH = 0 : 'Set illness and injury flags to normal
360 M = INT(M) : MA = M : 'Update cumulative mileage
370 IF MP = 1 THEN PRINT "Total mileage to date is 950." : MP = 0 : GOTO 400
380 PRINT "Total mileage to date is" INT(M + .5)
390 M = M + 200 + (A - 110) / 2.5 + 10 * RND(1) : 'Calculate how far we travel in 2 weeks
400 PRINT "Here's what you now have (no. of bullets, $ worth of other items) : "
410 GOSUB 3350 : 'Print inventory
420 GOSUB 1000 : 'Stop at fort, hunt, or push on routine
430 GOSUB 1310 : 'Eating routine
440 GOSUB 1390 : PRINT : 'Riders - attack routine
450 GOSUB 1800 : PRINT : 'Hazards - and - events routine

```

```

460 GOSUB 2640 : 'Mountains routine
470 GOTO 260 : 'End of the 2-week trip segment
480 '
490 'Subroutine to print initial scenario
500 PRINT TAB(23) "Westward Ho! 1847" : PRINT
510 PRINT " Your journey over the Oregon Trail takes place in 1847. Start."
520 PRINT "ing in Independence, Missouri, you plan to take your family of"
530 PRINT "five over 2040 tough miles to Oregon City."
540 PRINT " Having saved $420 for the trip, you bought a wagon for $70 and"

550 PRINT "now have to purchase the following items :" : PRINT
560 PRINT " * Oxen (spending more will buy you a larger and better team which"
570 PRINT " will be faster so you'll be on the trail for less time)"
580 PRINT " * Food (you'll need ample food to keep up your strength and health)"
590 PRINT " * Ammunition ($1 buys a belt of 50 bullets. You'll need ammo for"
600 PRINT " hunting and for fighting off attacks by bandits and animals)"
610 PRINT " * Clothing (you'll need warm clothes, especially when you hit the"
620 PRINT " snow and freezing weather in the mountains)"
630 PRINT " * Other supplies (includes medicine, first-aid supplies, tools, and"
640 PRINT " wagon parts for unexpected emergencies)" : PRINT
650 PRINT " You can spend all your money at the start or save some to spend"
660 PRINT "at forts along the way. However, items cost more at the forts. You"
670 PRINT "can also hunt for food if you run low." : PRINT : RETURN
680 '
690 'Subroutine to get initial purchases of player
700 PRINT : INPUT "How much do you want to pay for a team of oxen";A : A = INT(A)
710 IF A < 100 THEN PRINT "No one in town has a team that cheap." : GOTO 700
720 IF A < 151 THEN 760
730 PRINT "You choose an honest dealer who tells you that $" A "is too much for"
740 PRINT "a team of oxen. He charges you $150 and gives you $" A.150 "change."
750 A = 150
760 INPUT "How much do you want to spend on food";F : F = INT(F)
770 IF F > 13 THEN 790 ELSE PRINT "That won't even get you to the Kansas River";
780 PRINT " - better spend a bit more." : GOTO 760
790 IF A + F > 300 THEN PRINT "You won't have any for ammo and clothes." : GOTO 760
800 INPUT "How much do you want to spend on ammunition";B : B = INT(B)
810 IF B < 2 THEN PRINT "Better take a bit just for protection." : GOTO 800
820 IF A + F + B > 320 THEN PRINT "That won't leave any money for clothes." : GOTO 800
830 INPUT "How much do you want to spend on clothes";C : C = INT(C)
840 IF C > 24 THEN 860 ELSE PRINT "Your family is going to be mighty cold in";
850 PRINT " the mountains." : PRINT "Better spend a bit more." : GOTO 830
860 IF A + F + B + C > 345 THEN PRINT "That leaves nothing for medicine." : GOTO 830
870 INPUT "How much for medicine, bandages, repair parts, etc.";R
880 R = INT(R) : IF R < 5 THEN PRINT "That's not at all wise." : GOTO 870
890 IF A + F + B + C + R > 350 THEN PRINT "You don't have that much money." : GOTO 870
900 T = 350 - A - F - B - C - R : PRINT : PRINT "You now have $" T "left." : B = 50 * B :
RETURN
910 '
920 'Subroutine to initialize shooting routine
930 PRINT : PRINT "Please rank your shooting (typing) ability as follows :"
940 PRINT " (1) Ace marksman (2) Good shot (3) Fair to middlin'"
950 PRINT " (4) Need more practice (5) Shaky knees"
960 INPUT "How do you rank yourself";DR
970 IF DR > 1 AND DR < 6 THEN RETURN
980 PRINT "Please enter 1, 2, 3, 4, or 5." : GOTO 960
990 '
1000 'Subroutine to stop at fort, hunt, or push on
1010 IF INT(J / 2) < > J / 2 THEN 1060 : 'Are we on an even trip segment?
1020 INPUT "Want to (1) stop at next fort, (2) hunt, or (3) push on";X
1030 IF X < 1 OR X > 3 THEN PRINT "Enter a 1, 2, or 3 please." : GOTO 1020
1040 IF X = 3 THEN RETURN ELSE ON X GOSUB 1100, 1200
1050 IF KQ = 1 THEN 1020 ELSE RETURN : 'Not enough ammo to hunt?
1060 INPUT "Would you like to (1) hunt or (2) continue on";X
1070 IF X < 1 OR X > 2 THEN PRINT "Enter a 1 or 2 please." : GOTO 1060
1080 IF X = 2 THEN RETURN ELSE GOSUB 1200 : RETURN
1090 '
1100 'Subroutine to stop at a fort
1110 IF T > 0 THEN 1130 ELSE PRINT "You sing with the folks there and get a good"
1120 PRINT "night's sleep, but you have no money to buy anything." : RETURN
1130 PRINT "What would you like to spend on each of the following;"
1140 INPUT "Food";P1 : INPUT "Ammunition";P2 : INPUT "Clothing";P3

```



```

1150 INPUT "Medicine and supplies";P4 : P = P1 + P2 + P3 + P4 : P1 = .67 * P1 : P2 = 33 * P2
1160 PRINT "The storekeeper tallies up your bill. It comes to $" P
1170 IF T >= P THEN T = T - P : F = F + P1 : B = B + P2 : C = C + .67 * P3 : R = R + .67 * P4
: RETURN
1180 PRINT "Uh, oh. That's more than you have. Better start over." GOTO 1130
1190 '
1200 'Subroutine to hunt
1210 KQ = 0 : IF B > 39 THEN 1230 : 'Enough ammo to hunt?
1220 PRINT "Tough luck. You don't have enough ammo to hunt." : KQ = 1 : RETURN
1230 M = M - 45 : GOSUB 3870 : IF BR <= 1 THEN 1270
1240 IF 100 * RND(1) < 13 * BR THEN 1290
1250 PRINT "Nice shot...right on target...good eatin' tonight!"
1260 F = F + 24 - 2 * BR : B = B - 10 - 3 * BR : RETURN
1270 PRINT "Right between the eyes...you got a big one!" : F = F + 26 + 3 * RND(1)
1280 PRINT "Full bellies tonight!" : B = B - 10 - 4 * RND(1) : RETURN
1290 PRINT "You missed completely...and your dinner got away." : RETURN
1300 '
1310 'Subroutine to eat
1320 IF F < 5 THEN 3000 : 'Not enough food?
1330 INPUT "Do you want to eat (1) poorly (2) moderately or (3) well";E
1340 IF E < 1 OR E > 3 THEN PRINT "Enter a 1, 2, or 3, please." : GOTO 1330
1350 F = F - 4 - 2.5 * E : IF F > 0 THEN RETURN : 'Eating more food than you have?
1360 IF E = 1 THEN RETURN
1370 F = F + 4 + 2.5 * E : PRINT "You don't have enough to eat that well." : GOTO 1330
1380 '
1390 'Subroutine for riders attack
1400 IF RND(1) * 10 > ((M / 100 - 4) ^ 2 + 72) / ((M / 100 - 4) ^ 2 + 12) - 1 THEN RETURN
1410 X$ = "" : GH = 0 : IF RND(1) > .2 THEN X$ = "don't " : GH = 1
1420 PRINT : PRINT "Riders ahead! They " X$ "look hostile."
1430 PRINT "You can (1) run, (2) attack, (3) ignore them, or (4) circle wagons."
1440 INPUT "What do you want to do";GT
1450 IF GT < 1 OR GT > 4 THEN PRINT "Please enter 1, 2, 3, or 4." : GOTO 1440
1460 IF RND(1) < .2 THEN GH = 1 - GH : 'Maybe they're hostile after all
1470 IF GH = 1 THEN 1680 : 'Are they friendly?
1480 ON GT GOTO 1510, 1540, 1610, 1650
1490 '
1500 'Try to run away
1510 M = M + 20 : R = R - 7 : B = B - 150 : A = A - 20 : GOTO 1730 : 'Lose stuff when you run
1520 '
1530 'Attack the riders
1540 GOSUB 3870 : B = B - BR * 40 - 80 : 'Firefight uses ammo
1550 IF BR <= 1 THEN PRINT "Nice shooting - you drove them off." : GOTO 1730
1560 IF BR <= 4 THEN PRINT "Kind of slow with your Colt .45." : GOTO 1730
1570 PRINT "Pretty slow on the draw, partner. You got a nasty flesh wound."
1580 KH = 1 : PRINT "You'll have to see the doc soon as you can." : GOTO 1730
1590 '
1600 'Ignore the riders
1610 IF RND(1) > .8 THEN PRINT "They did not attack. Whew!" : RETURN
1620 B = B - 150 : R = R - 7 : GOTO 1730
1630 '
1640 'Circle wagons
1650 GOSUB 3870 : B = B - BR * 30 - 80 : M = M - 25 : GOTO 1550
1660 '
1670 'Cost of each tactic if riders were friendly
1680 IF GT = 1 THEN M = M + 15 : A = A - 5 : GOTO 1730
1690 IF GT = 2 THEN M = M - 5 : B = B - 100 : GOTO 1730
1700 IF GT = 3 THEN 1730 ELSE M = M - 20
1710 '
1720 'Final messages about riders
1730 IF GH = 0 THEN 1750 : 'Were riders hostile?
1740 PRINT "Riders were friendly, but check for possible losses." : RETURN
1750 PRINT "Riders were hostile. Better check for losses!"
1760 IF B >= 0 THEN RETURN ELSE PRINT : GOSUB 3740 : PRINT "Oh, my gosh!";
1770 PRINT "They're coming back and you're out of ammo! Your dreams turn to"
1780 PRINT "dust as you and your family are massacred on the prairie." : GOTO 3110
1790 '
1800 'Subroutine to deal with hazards and special events
1810 RN = 100 * RND(1) : 'RN determines which event happens
1820 FOR I = 1 TO 15 : 'Iterate through possible events
1830 IF RN > EP(I) THEN NEXT I : I = 16 : 'If event happened, exit loop
1840 IF I > 8 THEN 1860

```

```

1850 ON I GOTO 1880, 1910, 1940, 1980, 2010, 2040, 2080, 2180
1860 ON I - 8 GOTO 2290, 2320, 2350, 2410, 2440, 2530, 2560, 2610
1870 '
1880 PRINT "Your wagon breaks down. It costs you time and supplies to fix it."
1890 M = M - 15 - 5 * RND(1) : R = R - 4 : RETURN
1900 '
1910 PRINT "An ox gores your leg. That slows you down for the rest of the trip."
1920 M = M - 25 : A = A - 10 : RETURN
1930 '
1940 PRINT "Bad luck...your daughter breaks her arm. You must stop and"
1950 PRINT "make a splint and sling with some of your medical supplies."
1960 M = M - 5 - 4 * RND(1) : R = R - 1 - 2 * RND(1) : RETURN
1970 '
1980 PRINT "An ox wanders off and you have to spend time looking for it."
1990 M = M - 17 : RETURN
2000 '
2010 PRINT "Your son gets lost and you spend half a day searching for him."
2020 M = M - 10 : RETURN
2030 '
2040 PRINT "Nothing but contaminated and stagnant water near the trail."
2050 PRINT "You lose time looking for a clean spring or creek."
2060 M = M - 2 - 10 * RND(1) : RETURN
2070 '
2080 IF M > 950 THEN 2130 : 'If in mountains, go to snow; otherwise rain
2090 PRINT "Heavy rains. Traveling is slow in the mud and you break your spare"
2100 PRINT "ox yoke using it to pry your wagon out of the mud. Worse yet, some"
2110 PRINT "of your ammo is damaged by the water."
2120 M = M - 5 - 10 * RND(1) : R = R - 7 : B = B - 400 : F = F - 5 : RETURN
2130 PRINT "Cold weather...Brrrrrrrr!...You ";
2140 IF C < 11 + 2 * RND(1) THEN PRINT "don't "; : C1 = 1
2150 PRINT "have enough clothing to keep warm."
2160 IF C1 = 0 THEN RETURN ELSE GOSUB 2880 : RETURN
2170 '
2180 PRINT "Bandits attacking!" : GOSUB 3870
2190 B = B - 20 * BR : IF B > 0 THEN 2220 ELSE T = T / 3
2200 PRINT "You try to drive them off but you run out of bullets."
2210 PRINT "They grab as much cash as they can find." : GOTO 2230
2220 IF BR <= 1 THEN 2260 : 'Good response time?
2230 PRINT "You get shot in the leg - "; : GOSUB 3740 : KH = 1
2240 PRINT "and they grab one of your oxen." : A = A - 10 : R = R - 2
2250 PRINT "Better have a doc look at your leg...and soon!" : RETURN
2260 PRINT "That was the quickest draw outside of Dodge City."
2270 PRINT "You got at least one and drove 'em off." : RETURN
2280 '
2290 PRINT "You have a fire in your wagon. Food and supplies are damaged."
2300 M = M - 15 : F = F - 20 : B = B - 400 : R = R - 2 * 6 * RND(1) : RETURN
2310 '
2320 PRINT "You lose your way in heavy fog. Time lost regaining the trail."
2330 M = M - 10 - 5 * RND(1) : RETURN
2340 '
2350 PRINT "You come upon a rattlesnake and before you are able to get your gun"
2360 PRINT "out, it bites you." : B = B - 10 : R = R - 2 : IF R < 0 THEN 2390
2370 PRINT "Fortunately, you acted quickly, sucked out the poison, and"
2380 PRINT "treated the wound. It is painful, but you'll survive." : RETURN
2390 PRINT "You have no medical supplies left, and you die of poison." : GOTO 3060
2400 '
2410 PRINT "Your wagon gets swamped fording a river; you lose food and clothes."
2420 M = M - 20 - 20 * RND(1) : F = F - 15 : C = C - 10 : RETURN
2430 '
2440 PRINT "You're sound asleep and you hear a noise...get up to investigate."
2450 GOSUB 3740 : PRINT "It's wild animals! They attack you!" : GOSUB 3870
2460 IF B > 39 THEN 2480 ELSE PRINT "You're almost out of ammo; can't reach more."
2470 PRINT "The wolves come at you biting and clawing." : KH = 1 : GOTO 3030
2480 IF BR > 2 THEN 2500
2490 PRINT "Nice shooting, pardner...They didn't get much." : RETURN
2500 PRINT "Kind of slow on the draw. The wolves got at your food and clothes."
2510 B = B - 20 * BR : C = C - 2 * BR : F = F - 4 * BR : RETURN
2520 '
2530 PRINT "You're caught in a fierce hailstorm; ammo and supplies are damaged."
2540 M = M - 5 - 10 * RND(1) : B = B - 150 : R = R - 2 - 2 * RND(1) : RETURN
2550 '

```

```

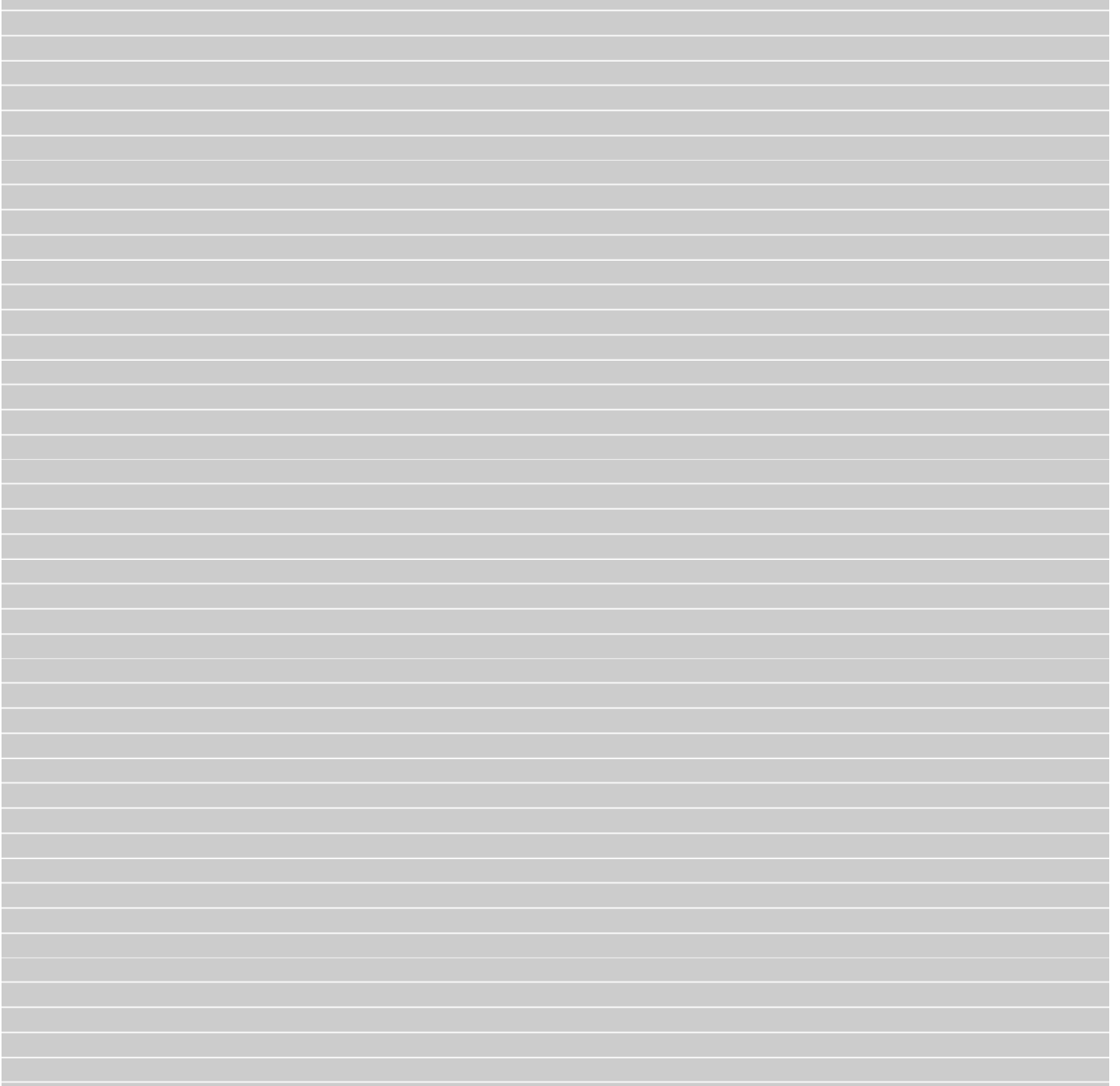
2560 'Problems from not eating well enough?
2570 IF E = 1 THEN GOSUB 2880 : RETURN : 'If eating poorly, go to sickness routine
2580 IF E = 2 AND RND(1) > .25 THEN GOSUB 2880 : RETURN
2590 IF E = 3 AND RND(1) > .5 THEN GOSUB 2880 : RETURN
2600 '
2610 PRINT "Helpful Indians show you where to find more food."
2620 F = F + 7 : RETURN
2630 '
2640 'Subroutine to travel through mountains
2650 IF M <= 975 THEN RETURN : 'Not in mountains yet?
2660 IF 10 * RND(1) > 9 - ((M / 100 - 15) ^ 2 + 72) / ((M / 100 - 15) ^ 2 + 12) THEN 2750
2670 PRINT "You're in rugged mountain country." : IF RND(1) > .1 THEN 2700
2680 PRINT "You get lost and lose valuable time trying to find the trail."
2690 M = M - 60 : GOTO 2750
2700 IF RND(1) > .11 THEN 2730
2710 PRINT "Trail cave in damages your wagon. You lose time and supplies."
2720 M = M - 20 - 30 * RND(1) : B = B - 200 : R = R - 3 : GOTO 2750
2730 PRINT "The going is really slow; oxen are very tired." : M = M - 45 - 50 * RND(1)
2740 '
2750 'South Pass routine
2760 IF KP = 1 THEN 2790 : 'Is the South Pass clear?
2770 KP = 1 : IF RND(1) < .8 THEN 2840 : '80% chance of blizzard
2780 PRINT "You made it safely through the South Pass....no snow!"
2790 IF M < 1700 THEN 2810
2800 IF KM = 1 THEN 2810 : 'Through Blue Mts yet?
2810 KM = 1 : IF RND(1) < .7 THEN 2840 ELSE RETURN : 'Get through without mishap?
2820 MP = 1 : RETURN : 'Set South Pass flag
2830 '
2840 PRINT "Blizzard in the mountain pass. Going is slow; supplies are lost."
2850 KB = 1 : M = M - 30 - 40 * RND(1) : F = F - 12 : B = B - 200 : R = R - 5
2860 IF C < 18 + 2 * RND(1) THEN GOTO 2880 ELSE RETURN : 'Enough clothes?
2870 '
2880 'Subroutine to deal with illness
2890 IF 100 * RND(1) < 10 + 35 * (E - 1) THEN 2930
2900 IF 100 * RND(1) < 100 - (40 / 4 ^ (E - 1)) THEN 2950
2910 PRINT "Serious illness in the family. You'll have to stop and see a doctor"
2920 PRINT "soon. For now, your medicine will work." : R = R - 5 : KS = 1 : GOTO 2970
2930 PRINT "Mild illness. Your own medicine will cure it."
2940 M = M - 5 : R = R - 1 : GOTO 2970
2950 PRINT "The whole family is sick. Your medicine will probably work okay."
2960 M = M . 5 : R = R . 2.5
2970 IF R > 0 THEN RETURN ELSE PRINT " ...if only you had enough." : GOTO 3020
2980 '
2990 'Many ways to die on the trail
3000 PRINT "You run out of food and starve to death." : GOTO 3110
3010 T = 0 : PRINT "You need a doctor badly, but can't afford one." : GOTO 3030
3020 PRINT "You have run out of all medical supplies."
3030 PRINT : PRINT "The wilderness is unforgiving and you die of ";
3040 IF KH = 1 THEN PRINT "your injuries." : GOTO 3060
3050 PRINT "pneumonia."
3060 PRINT "Your family tries to push on, but finds the going too rough";
3070 PRINT " without you." : GOTO 3110
3080 PRINT "Your oxen are worn out and can't go another step. You try pushing"
3090 PRINT "ahead on foot, but it is snowing heavily and everyone is exhausted."
3100 PRINT : GOSUB 3740 : PRINT "You stumble and can't get up...."
3110 PRINT : GOSUB 3740 : PRINT "Some travelers find the bodies of you and your"
3120 PRINT "family the following spring. They give you a decent"
3130 PRINT "burial and notify your next of kin." : PRINT
3140 D = INT(14 * (J + ML)) : DM = INT(D / 30.5) : DD = INT(D - 30.5 * DM)
3150 PRINT "At the time of your unfortunate demise, you had been on the trail"
3160 PRINT "for" DM "months and" DD "days and had covered" INT(M + 70) " miles."
3170 PRINT " You had a few supplies left :" : GOSUB 3350 : PRINT : GOTO 3310
3180 '
3190 'Made it!
3200 ML = (2040 - MA) / (M - MA) : F = F + (1 - ML) * (8 + 5 * E) : GOSUB 3830
3210 PRINT "You finally arrived at Oregon City after 2040 long miles."
3220 PRINT "You're exhausted and haggard, but you made it! A real pioneer!"
3230 D = INT(14 * (J + ML)) : DM = INT(D / 30.5) : DD = INT(D - 30.5 * DM)
3240 PRINT "You've been on the trail for" DM "months and" DD "days."
3250 PRINT "You have few supplies remaining :" : GOSUB 3350
3260 PRINT : PRINT "President James A. Polk sends you his heartiest"

```

```

3270 PRINT "congratulations and wishes you a prosperous life in your new home."
3280 GOTO 3310
3290 '
3300 'Play - again query
3310 PRINT : INPUT "Would you like to play again";A$ : GOSUB 3770
3320 IF A = 0 THEN PRINT "Okay, good luck!" : GOSUB 3740 : RUN
3330 PRINT "Okay. So long for now." : GOSUB 3740 : KEY ON : CLS : END
3340 '
3350 'Subroutine to print inventory
3360 PRINT "Cash", "Food", "Ammo", "Clothes", "Medicine, parts, etc."
3370 IF F < 0 THEN F = 0 ELSE F = INT(F)
3380 IF B < 0 THEN B = 0 ELSE B = INT(B)
3390 IF C < 0 THEN C = 0 ELSE C = INT(C)
3400 IF R < 0 THEN R = 0 ELSE R = INT(R)
3410 PRINT T, F, B, C, R : PRINT : RETURN
3420 '
3430 'Subroutine to read shooting words and dates
3440 FOR I = 1 TO 8 : READ S$(I) : NEXT I
3450 DATA "POW", "BANG", "BLAM", "WHOP", "pow", "bang", "blam", "whop"
3460 FOR I = 1 TO 20 : READ D$(I) : NEXT I : RETURN
3470 DATA "March 29", "April 12", "April 26", "May 10", "May 24", "June 7", "June 21"
3480 DATA "July 5", "July 19", "August 2", "August 16", "August 31", "September 13"
3490 DATA "September 27", "October 11", "October 25", "November 8", "November 22"
3500 DATA "December 6", "December 20"
3510 '
3520 'Subroutine to read distances and place names
3530 FOR I = 1 TO 15 : READ MP(I), PL$(I) : NEXT I : RETURN
3540 DATA 5, "on the high prairie."
3550 DATA 200, "near Independence Crossing on the Big Blue River."
3560 DATA 350, "following the Platte River."
3570 DATA 450, "near Fort Kearney."
3580 DATA 600, "following the North Platte River."
3590 DATA 750, "within sight of Chimney Rock."
3600 DATA 850, "near Fort Laramie."
3610 DATA 1000, "close upon Independence Rock."
3620 DATA 1050, "in the Big Horn Mountains."
3630 DATA 1150, "following the Green River."
3640 DATA 1250, "not too far from Fort Hall."
3650 DATA 1400, "following the Snake River."
3660 DATA 1550, "not far from Fort Boise."
3670 DATA 1850, "in the Blue Mountains."
3680 DATA 2040, "following the Columbia River."
3690 '
3700 'Read probabilities of events
3710 FOR I = 1 TO 15 : READ EP(I) : NEXT : RETURN
3720 DATA 6, 11, 13, 15, 17, 22, 32, 35, 37, 42, 44, 54, 64, 69, 95
3730 '
3740 'Subroutine to create a short pause
3750 FOR I = 1 TO 1000 : NEXT I : RETURN
3760 '
3770 'Subroutine to read a yes/no answer
3780 X$ = LEFT$(A$, 1) : IF X$ = "Y" OR X$ = "y" THEN A = 0 : RETURN
3790 IF X$ = "N" OR X$ = "n" THEN A = 1 : RETURN
3800 PRINT "Don't understand your answer of " A$ "."
3810 INPUT "Please enter Y for 'yes' or N for 'no.' Which is it";A$ : GOTO 3780
3820 '
3830 'Subroutine to play a fanfare
3840 FQ = 220 : SOUND FQ, 3 : SOUND FQ * 1.333, 3 : SOUND FQ * 1.584, 3
3850 SOUND FQ * 1.885, 5 : SOUND FQ * 1.584, 2 : SOUND FQ * 1.885, 10 : RETURN
3860 '
3870 'Subroutine to shoot gun
3880 RN = 1 + INT(4 * RND(1)) : 'Pick a random shooting word
3890 S1 = 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) : 'Start timer
3900 PRINT "Type " S$(RN); : INPUT X$
3910 IF S$(RN) < >X$ AND S$(RN + 4) < >X$ THEN PRINT "Nope. Try again. "; : GOTO 3900
3920 S2 = 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) : 'End timer
3930 BR = S2 - S1 - DR - 1 : RETURN

```





THE GAME

The Longest Automobile Race puts you in the driver's seat. In this simulation of a 1908 race from New York to Paris—across the U.S., Japan, Russia, and Europe—you are the captain of the American team. It is your job to get the Thomas Flyer to Paris as quickly as possible—and before your money runs out.

Soon after your departure on February 12, 1908, you encounter some of the worst snows of the decade. From Chicago on, you face a different array of problems: accidents, fatigue, mechanical breakdowns, and more bad weather. Japan and Russia have their own problems (narrow cart tracks in Japan, no gas or lubricants in Russia).

Your chances of beating the other contestants to Paris will be increased if you note the following:

- Buy as much gas as you need, but no more. Prices are high between stops, and any gas remaining at the end of a segment goes to waste. Your car gets approximately 14 miles per gallon, but with changing terrain your mileage could vary by 20%.
- At speeds in excess of 35 mph, the probability of mechanical breakdown increases dramatically. Likewise, driving more than six and a half hours per day increases your chances of having an accident.
- If you get stuck in a snowdrift, a ditch, or mud, you can either pay someone to pull you out or try to extricate yourself. The former costs you money; the latter, time.
- You can choose whether to repair a mechanical problem on the spot or nurse the car along to the next city. On-the-spot repairs are generally more costly, but a second malfunction on top of the first forces you to retire from the race.

On July 30, 1908, nearly six months after starting the race, the weary Thomas Flyer crew limped down the Champs-Elysees and was declared the winner of the race. Can you improve on the performance of the American team—or will your opponents pass as you are permanently bogged down in Siberian mud?

THE STORY

In 1908, barely 15 years after the automobile was invented, the seemingly preposterous suggestion that an automobile race be run from New York to Paris was made. This race, proposed by newspapermen of the *New York Times* and *Le Matin* of Paris, was to be made over untried roads; the sponsors believed it would be possible to drive in Alaska by widening dog trails and, in the far North, driving on the crust of the snow that, according to one *Times* document, "froze so solidly overnight that it would bear the weight of a car." They also believed that contestants would be able to drive a car on frozen rivers and to cross the Bering Strait between Alaska and Siberia on an ice bridge.

This sort of misinformation, coupled with the daring, adventuresome spirit of early automobile makers and the promise of honor and glory, lured thirteen contestants to sign up for the race.

However, on February 12, 1908, the day the race was to begin, only six teams actually showed up. The entrants included three cars made in France and one each made in Italy, Germany, and the United States.

The French entries included a 30 hp Motobloc, a big De Dion, and a tiny one-cylinder, 15 hp Sizaire-Naudin. Italy was represented by a Zust, Germany by a big Protos, and the U.S. by a year-old, 60 hp Thomas Flyer. All of the cars had four cylinders (except the Sizaire-Naudin), open bodies with no windshields, narrow spoked wheels, no heaters, and only the most basic seating for driver and passengers.

Each team consisted of a driver and one or more mechanics. In addition, each car was required to carry a writer/observer from one of the sponsoring newspapers.

The race got underway in front of the Times Building on 43rd Street at 11:15 A.M. on Abraham Lincoln's birthday, 1908.

The Sizaire-Naudin driven by Auguste Pons led the procession up Broadway and onto the Albany Post Road. It was the only time Pons had the lead; while climbing a hill only 40 miles from the city, his car broke an axle. There were no spare parts available for his car, and Pons spoke no English. Thus the race ended for him on the very first day.

Language was also a problem for the crew of the French Motobloc. They stopped for lunch at a hotel in Dobbs Ferry, New York, and after waiting two hours to be served, became furious and accused the proprietor of overcharging and delaying them. To add injury to insult, just a few miles north at Peekskill, the Motobloc skidded in the snow and landed in a ditch, delaying it further. Snow and language difficulties continued to plague the Motobloc team, and they finally dropped out of the race three weeks later in western Iowa.

The remaining four cars fought on—although their battles were as much or more with Mother Nature as with one another.

From the newspaper stories, one can piece together a fascinating description of the race. However, a somewhat more coherent account is found in the book, *The Longest Auto Race*, by George Schuster, the mechanic on the American Thomas Flyer. Moreover, Schuster and the Flyer team stayed in the race long after even the hardest newspapermen had dropped out.

Schuster records constant problems with snow as the cars progressed along the route from New York to Albany, Buffalo, Erie, Cleveland, South Bend, and Chicago. In fact, Ohio and Indiana had one of the worst blizzards of the decade just after the cars left Buffalo.

What should have been a two- or three-day trip from Buffalo to Chicago became a week-long ordeal for the Flyer team: in and out of snow drifts, driving on railroad tracks, hiring teams of horses to pull the car a few miles, and

dealing with every sort of mechanical problem.

Beyond South Bend, the team was forced to hire teams of six or eight horses to drag the car along. On February 22 Schuster spent \$80 on horse teams and covered only 15 miles. The next day, the Flyer team struggled for 22 hours and covered a grand total of eight miles.

The chores of draining the water from the cooling system every night (antifreeze had not yet been invented) and making constant small repairs added to the frustration of slow progress.

Finally, at 4:25 P.M. on February 25, the Flyer team reached the South Shore Country Club, the official arrival point in Chicago. Schuster notes that it took 13 1/2 days to travel the 1403 miles from New York, eight days of which were required to cover the last 256 miles. All of the teams spent a few days in Chicago recuperating and making needed repairs before starting westward on "tracks of half-frozen ruts" on the morning of February 28.

Now the battle was with thick, gooey mud, which the wheels and tires threw in big gobs onto the fenders, running boards, and interior of the car. Schuster records that when the Flyer team pulled into Clarence, Iowa, to stop for the night, someone suggested that they take the car to the fire station to get the mud washed off. The high-pressure three-inch stream was so effective that the Flyer crew visited fire stations whenever they could from then on. The thrill of the trip had worn thin for the *Times* correspondent, and a day later in Cedar Rapids he bade the American crew good bye and returned to New York.

The Plains States brought more rain and mud, which, as the cars reached the mountains, turned into sleet, hail, and snow.

Conditions were so bad in Utah and eastern Nevada that it became clear that crossing the Sierra Nevada Mountains (the direct route to San Francisco) would be impossible. So the Flyer team took a 700-mile detour south through the Nevada desert and Death Valley to Bakersfield, California and back up along the San Joaquin Valley to San Francisco.

The Flyer team made it to San Francisco on March 24, a total of 3836 miles and 41 days after leaving New York. At the time, the Italian Zusta was in Utah and the French De Dion and the German Protos were in Wyoming, all pushing west. With the help of the Thomas Flyer dealer in San Francisco, that car was prepared, as much as possible, for the unknown rigors of Alaska and Siberia that lay ahead. Springs, transmission, wheels, drive chains, and many other parts were replaced in the three days before the car was loaded on the steamer City of Pueblo for the three-day voyage to Seattle. There, five tires—all that could be found—were purchased and Schuster wired owner Mr. Thomas at the factory for additional funds.

The crew and car sailed north on a leisurely seven-day voyage to Valdez, Alaska. There they learned the truth about the proposed route: Dog trails didn't exist, snow would not support a car, and there was no ice bridge across the Bering Strait. They telegraphed this information to New York and soon received a response: "Return to Seattle. Route changed. Go by steamer to Vladivostok." Meanwhile, the Zusta and the De Dion had arrived in San Francisco, while the Protos, much in need of repairs, had been shipped by its crew aboard a railroad flatcar from Idaho across the Sierra Nevadas to Seattle, avoiding the 700-mile "detour" taken by the other cars.

The Protos sailed immediately for Vladivostok; the other teams, unable to get a steamer for Russia, sailed instead for Japan. The Flyer crew, afraid that those teams would get credit for driving across Japan, decided to take a steamer for Kobe, Japan. On the voyage, the leather aprons with which the Flyer crew had replaced their metal fenders were removed by the Chinese crew to make sandals. The captain ordered the ship's carpenter to replace the aprons with canvas and, Schuster reported, "except for that our voyage was uneventful." As it turned out, the Flyer crew need not have worried about the newly added Japan segment. The race committee ruled that the Protos would be penalized 15 days for not driving all the way across the U.S. and that the Flyer crew would be given a 15-day advantage for going to Alaska.

The trip across Japan from Kobe to Tsuruga was only 350 miles, but torturous miles they were. Japanese roads were designed for narrow carts, and the big cars could barely negotiate the sharp turns and steep hills. What should have been an easy two-day drive turned into a four-day ordeal punctuated by delays for religious processions, detours, wrong turns, and skids into ditches. Although it was commonly thought that the Japanese worked for coolie wages, one group of workmen insisted on 50 yen (\$25) for manning a towrope to let the Flyer down a steep hillside.

Sailing from Tsuruga on May 16, the Flyer and its crew finally arrived in Vladivostok in a gloomy rain on May 18. The other cars were all there, but the eccentric Count De Dion, his team having already driven and won the

Peking to Paris race the year before, decided he had much to lose and little to gain by continuing on, and pulled his team out of the race.

Gasoline was extremely scarce in Russia, and the Protos and Zust teams had lined up most of the existing supplies. Schuster was thus forced to cash his last letter of credit to buy 400 gallons of gasoline at exorbitant prices (\$1.00 to \$1.25 per gallon). Most of this was then shipped ahead by rail to be stored along the route where the team could retrieve it in the days ahead. This was necessary because there were no filling stations, and gasoline would not be commercially available until the cars reached Europe. The teams stocked up on food and supplies and were ready to go on May 22. "You are mad," a Russian officer told them. "You will never get through." It rained during 17 of the next 20 days and, recounted one driver, "we drove out into a dismal, flat, rain-drenched country, over or rather through a road that was a streak of mud as far as the eye could reach." The Flyer drove in low or second gear with chains on the rear wheels.

In one of the most memorable events of the race, the Flyer came upon the Protos so deep in the mud that only the tops of the rear wheels showed above the mire. The three Germans and a Russian officer/guide were trying to pry it out, but with the churning of the wheels it was only sinking deeper. After a brief debate among themselves, the Flyer crew passed a towrope to the Germans and managed to pull the Protos to solid ground. Lt. Koeppen, the German team captain, uncorked a bottle of champagne and poured drinks as thanks for what he called "a gallant, comradely act."

Finally, unable to make any progress in the muddy cart tracks, the Flyer crew took to driving along railroad tracks, a strategy they had successfully used—at great cost to the wheels, tires, and springs—on several stretches in the U.S. Two tires quickly blew out, and four more were worn to the cords in the next 150 miles. The organizers had ruled that the cars would not be permitted to fit special wheel flanges that would allow them to drive on the rails but that driving on the ties was permitted.

Soon, however, the pounding on the ties began to take an even greater toll than the tires and springs. A sharp noise signaled a six-inch crack in the Flyer transmission housing and the stripping of six gear teeth from the drive pinion. Schuster then made a five-day trip to Harbin where he cabled the factory to send a new transmission by way of Europe. Meanwhile, the crew made makeshift repairs—new teeth were welded to the drive pinion at a blacksmith shop—and slowly pushed on.

Mechanical devices, much less automobiles, were virtually unknown in Manchuria and Siberia, and lubricants were unavailable. Thus the crew was forced to buy 40 lbs. of Vasoline in Chita to quiet the gears and allow them to keep moving. At Irkutsk, completely out of money, Schuster again wired the factory for more. He received it, but Thomas warned him that he was disinclined to pour any more money into this adventure. Also at Irkutsk when the mud was washed from the car, he discovered a broken motor support. Schuster replaced it with a piece of boiler plate from the railroad shops, but the break had thrown the motor out of line, making it difficult to shift gears.

It rained so constantly through Siberia that the road was a continuous bog. The car frequently got stuck, and laborers or horse teams had to be hired to pull it out. Near Kansk was a swollen river with a quicksand bottom and no bridge; there the Flyer crew hired four teams of Cossacks' horses to rush the car across a fording spot. The next day they came upon a river so deep it could not be forded, so the crew hired villagers to make a log raft to float the car across.

Deep ditches across the roads were frequent hazards in Siberia. On one occasion, completely worn brake drums failed to stop the Flyer in time at one of these ditches; there was a sudden snap, and the left side of the car sagged. The frame was broken, creating an awkward problem in the middle of nowhere. Finally, angle iron was secured from a railroad shop and bolted to the broken frame.

The Protos had been in the lead across most of Russia, but finally, near the village of Kolnokowa, the Flyer caught up with and passed the German competitor. However, a few days later, on July 1 in Omsk, 3408 miles from Vladivostock, the makeshift transmission repairs on the Flyer gave out. The new transmission had been shipped two weeks earlier aboard the SS President and should have been waiting. It wasn't, so again the crew laboriously forged new teeth at a blacksmith shop and welded them on. This cost three days, but on July 4, the Flyer was back on the road. The Protos, too, had run into trouble and was obliged to wait in Kansk for five days until a new rear axle could be shipped from the factory in Germany.

Schuster, fearing that the transmission would soon fail permanently, sent two crew members (two mechanics from teams that had dropped out had joined the Flyer crew) to check all the railroad depots in the area for the missing transmission while the car pushed slowly on toward Perm. There he received word that the missing transmission had been located at Kasan, 350 miles ahead and more or less on the route west. However, the mud was so bad

that the car could never be shifted out of low, and three days out of Perm—still 215 miles from Kasan—the transmission failed for good.

Schuster then used the last of his money to arrange relays of farm wagons to fetch the 600-lb. transmission. It took five days, by which time Schuster, walking with the wagons day and night, was sick with chills and a high fever. Installing the transmission took another day, and on July 18 the team finally reached Kasan where they took a steam ferry across the Volga River. A wrong turn, a leaky radiator, and another break in the frame slowed the Flyer, but finally at Gorky they reached the relatively smooth pavement of some of the oldest roads in Europe.

But there were problems with these roads, too. When driven at speed on a good surface, the Flyer wobbled all over the road; the repaired motor support broke again and was out of alignment with the clutch and transmission. The tires were worn through, and the radiator was still leaking. Thus, the team had to stop for several days in Moscow to make more repairs.

The Flyer team reached St. Petersburg on the afternoon of July 22, four days behind the Protos; three days later, the Flyer crossed the border into Germany. The radiator sprang yet another leak and was removed for more repairs at the ancient fortress city of Königsberg. The crew reached Berlin on July 27, and there heard the disheartening news that the Protos had arrived in Paris the night before. Rolling along perfectly smooth road outside of Hanover, the car slowed down and finally coasted to a stop. The makeshift clutch shaft installed in Moscow had worn away completely, again necessitating complete disassembly and repair. The 30-lb. assembly was hauled to a machine shop many miles away where it took another half day to make and install a new part.

At noon the next day, having had little food or rest, the Flyer crew drove into Cologne, looking forward to a washup and a good meal. "Not here, not here," said the head waiter of the Dom Hotel, shaking his head and waving the motley crew out of the restaurant. After an unsatisfying meal at a small cafe, they left the city and crossed the Rhine into Liege, Belgium, to spend the night.

Leaving early on July 30, they followed the Meuse River and crossed into France at Fumay. Accelerator to the floor, they attained a speed of 50 mph, the highest since they had left the United States. Approaching Paris late in the afternoon, they were stopped by gendarmes who refused to let them pass without headlamps (they had broken weeks before in Siberia). A passing Frenchman gallantly offered one from his bicycle. When the lamp could not be detached, the bicycle was lifted onto the car, and the journey was resumed amid cheering crowds.

At 6:00 P.M. on July 30, the Thomas Flyer pulled up in front of the offices of *Le Matin* and, with the credit for going to Alaska and the penalty against the Protos, was declared the winner of the longest automobile race in history by 26 days. The only other car still in the race, the Italian Zusta, finally reached Paris on September 17, missing second place by only three days.

Winning the race briefly spurred sales of the Thomas Flyer, but the euphoria was short-lived. In 1909, Henry Ford introduced the Model T, thus ending the age of expensive hand-crafted automobiles. A year later, only 913 Thomas Flyers were produced, and the following year the company left the business.

Note: The restored chassis of the competing Protos is on display in the Deutsches Museum in Munich, Germany, and the restored Thomas Flyer is at the Harrah Automotive Museum in Reno, Nevada. The Zusta was not preserved.

BIBLIOGRAPHY

Schuster, George with Tom Mahoney. *The Longest Auto Race*. New York: The John Day Co., 1966.

The New York Times. Various accounts between February and August, 1908.

THE PROGRAM

The Longest Automobile Race program consists of a main program, initialization section, summary section, 12 major subroutines, and five lesser subroutines.

In the initialization section (Lines 140–230), variables are dimensioned, and text and numeric data are read into variables in three subroutines. When the data are loaded, the program enters an endless loop (Line 220) to produce a seed for the Randomize function. This loop is exited when you press any key.

The main program (Lines 250–620) consists of two main sections. The first section (Lines 260–500) obtains data for the next section of the race, specifically weather (W), distance (D), and the number of days that it took the winning car to complete the segment in 1908 (TE). The current date, location, and summary information of the

race to that point are then printed. If the racer has managed to limp along with an unfixed problem that developed on the road, repairs are made at this location (Lines 310–330).

If the next segment of the race is an ocean voyage, the ocean voyage subroutine (Lines 2290–2570) is called. If the next race segment is a normal land segment, the main program prints a description of the expected weather and road conditions, and establishes a time and distance goal (Lines 450–470).

Next, three subroutines that accept input from the user are called: the amount of fuel to be put in the car and spotted at villages ahead (in many locations, the racing teams had to send cans of gasoline ahead by train), the desired driving speed, and the desired number of driving hours per day.

The second major section of the main program (Lines 520–620) loops through each day of the race. For each day, subroutines that deal with weather conditions, mechanical breakdowns, and accidents and special situations, are called. After this, the program calculates daily and cumulative distance traveled and gasoline consumption. If a race segment is completed, the program branches back to the beginning of the main program to start the next segment.

In the gas-and-oil subroutine (Lines 640–710), you are asked how many gallons of fuel you want to purchase. The price of gasoline varies between 17 and 39 cents per gallon (Line 650); the average is 25 cents. However, if you run out of gasoline on the road, the base price is 33 cents per gallon, although it can be as low as 23 cents or as high as 52 cents. In general, the best strategy is to buy enough gas for a segment at the outset. However, extra gas goes to waste, as it cannot be shipped ahead to the next race segment. In general, the car will get approximately 14 miles per gallon, but mileage varies by plus or minus 20% (Line 580). After calculating the total cost of the gas you want to buy, the program goes to the payment subroutine at Line 2660 (discussed later). If you don't have enough money to buy the amount of gasoline you want, you get as much as you can afford.

The subroutine to accept the desired speed (Lines 730–800) will accept any speed between 8 mph and 54 mph, the top speed of the Thomas Flyer. In snowy weather, the program limits the maximum speed to 30 mph. Of course, your average speed on the road will be lower than the speed you input, because you must slow down to go through villages, make detours, make rest stops, and deal with unexpected problems. The probability of a mechanical breakdown is related to speed by the formula at Line 790 and increases rather quickly as speed exceeds 35 mph. You must, of course, balance the risk of mechanical problems with that of driving too slowly to be competitive.

The driving-hours subroutine (Lines 820–920) accepts the desired driving hours per day (HP), establishing an upper limit of eight and a lower limit of two. If you have consistently been driving for an average of more than seven and a half hours, you are advised not to push yourself and your crew quite so hard. The probability of a problem resulting from driver fatigue (PF) is related to driving hours and starts to rise sharply when six and a half hours per day is exceeded (Line 910). The formula for this probability is:

$$PF = HP^3 / 1000 - .15$$

Cubing the hours causes the very sharp rise from nil at five hours per day to 6.6% at six hours, 19.3% at seven hours, and 36.2 at eight hours.

The date subroutine (Lines 940–1100) simply takes the day counter (TD%XC) and converts it into the proper month and day. If the date goes beyond August 31—one full month after the first car crossed the finish line in 1908—you are obliged to withdraw from the contest.

The weather subroutine (Lines 1120–1620) is the longest in the program. It is divided into five sections, representing major types of weather conditions, and a subroutine having to do with getting stuck in a ditch.

Each of the five major weather-subroutine sections is structured similarly. First, a random number is generated to determine if something nasty happens. For example, in the heavy snow section (Lines 1150–1210), there is a 33% chance of hitting a blizzard of major proportions. In this case the speed factor (PW) ranges from .03 to .10; i.e., your actual speed will be only 3% to 10% of what you input. You also have a 17% chance of getting stuck in a snowdrift, in which case you are sent to the stuck-in-the-ditch subroutine. Under "normal" heavy-snow conditions, the speed factor is about .22 or 22% of what you input.

In regular snowy conditions there is a 10% chance of skidding off the road, but in all cases your speed is much reduced (Lines 1230–1270).

Rainy weather was a much greater problem in 1908 than it is today, because the dirt roads turned into virtual quagmires into which a car could sink 10 or 15 inches. There is a 20% chance of this occurring (Line 1300).

Furthermore, if you do get stuck, getting out is usually rather time-consuming.

In the cloudy-and-mixed-weather section (Lines 1350–1470), there is a 1% chance of a sudden downpour, an 8% chance of coming to a river with no bridge, and a 92% chance of smooth sailing. Although a river without a bridge doesn't seem to be related to cloudy weather, in the actual 1908 race most of the rivers across which the contestants had to be ferried were reached during stretches of cloudy or mixed weather. The program gives you a choice of being ferried across (for a fee) or driving north to a bridge (an unknown distance away).

If you get stuck in a snowdrift, a ditch, or mud, you can usually find a farmer to pull you out for \$5 to \$20 (Line 1540). If you want to pay him to pull you out, the program calls the payment routine. If not, you and your mechanic have at it with prybars, shovels, and a winch. To get out by yourself takes at least one day and sometimes two (Lines 1610–1620).

The mechanical-breakdowns subroutine (Lines 1640–1910) is perhaps the most complex in the program. First, a random number is compared to the breakdown probability to determine if there actually is a breakdown (Line 1650). Then, the specific problem is determined by the random number functions in Line 1660; the first 13 malfunctions are twice as likely to occur as the last five.

```
1660 F = INT(1 + 15 * RND(1)) : IF F > 13 THEN F = INT(14 + 5 * RND(1))
```

The problem is then presented to you, and you are asked whether or not you want to fix it and which type of repair you want to use in the event that more than one is available. One is usually a new part and the other a homebrew approach. Each repair has a time and cost associated with it. If you have enough money, the repairs are made; otherwise the car must be nursed along to the next city at half speed, a procedure with a 60% chance of success. Also, if there is a second unrepaired malfunction, the combination of the two forces the car to be retired. In general, the only time you might not want to do a repair right away is when you are nearing the next city, because you won't lose much time at the reduced speed and also a sympathetic automobile dealer will usually fix your car for free (Lines 310–330).

The accidents-and-special-situations subroutine (Lines 1930–2270) is divided into three sections. The first determines whether or not there is an accident (Line 1940). As mentioned earlier, the probability of an accident is related to the average driving hours per day. If there is an accident—you dozed off and ran into something: a tree, hole, or farmer's wagon—you can try to fix it on the spot (costs time) or get a tow to the next village and get it repaired there (costs money). At the time of the accident, the farmer will tell you the cost of a tow to the next village, but, of course, you will not know how much the repairs will cost.

In the four race segments in which the roads are particularly bad or blocked with snow, you may be able to get permission to drive on the railroad tracks (Lines 2100–2170). While this will save you time—the speed factor is multiplied by 1.7—the constant pounding of tires on the uneven railroad ties is extremely hard on the car.

Although the wheels, tires, suspension, transmission, and chassis take the biggest beating, in the program the current overall probability of all problems is simply multiplied by 1.25, i.e., an increase of 25% (Line 2170).

In the two race segments in central Russia between Chita and Omsk, very few mechanical supplies of any sort were available in 1908. Gasoline, oil, grease, and all spare parts had to be carried on the cars or shipped ahead, or substitutions made on the spot. During this segment, the Thomas Flyer developed a serious crack in its transmission. The team requested that a replacement be shipped from the factory, but they had to push on to meet the transmission in Perm.

A portion of this adventure is simulated in the no-grease routine (Lines 2190–2270). Also, the time for a transmission repair is set to forty hours (Line 3510), which is five eight-hour workdays. The forty hours is converted to days in Line 1790 so it can be added to the cumulative travel time.

The ocean-voyage subroutine (Lines 2290–2570) simulates the five steamer voyages: San Francisco to Seattle, Seattle to Valdez, Valdez to Seattle, the long 21-day trip to Kobe, Japan, and the short voyage from Tsuruga to Vladivostok, Russia. There is not much decision-making on these segments. Your waiting time in port varies from one to four days and is produced by the formula in Line 2300.

A short subroutine takes care of time delays and hotel stays (Lines 2590–2640). It simply increments the day counters by TZ, a variable set in various other subroutines representing days needed for repairs, tows, river crossings, and port stays. For each day of the trip, except those on ships, you pay an average of \$10 for food and accommodations for you and your mechanic.

All bills—repairs, tows, hotels, gasoline, and ferries—are paid in a four-line subroutine (Lines 2660–2690). The cost of an item (ZN) is simply subtracted from your cash reserve (Z). If you don't have enough money to pay your bills and can't get it from home, an indicator (A) is set to 1 and program execution is returned to the place the bill was incurred. The consequences of not paying your bills can range from mild (if you can't pay for as much gasoline as you want, you are given as much as you can afford) to severe (unpaid repair and hotel bills take you out of the race).

If you run out of money, you can wire Mr. Thomas at the factory in Buffalo for more (Lines 2710–2950). In 1908, Mr. Thomas was wildly enthusiastic about the race before it started and while the cars were in the U.S. However, as the months wore on his enthusiasm turned to disenchantment, and he became much less willing to supply additional funds to the crew. The growing disillusionment of Mr. Thomas as well as his somewhat temperamental nature is simulated in the program.

You can go back and ask for money three times before Mr. Thomas gets totally fed up. You can get up to \$1000 on each of the first two requests and up to \$500 on the third. But if you don't say the right things you may get less or you may have to wait for the money. Lines 2790 to 2830 pick apart the contents of your telegram to Mr. Thomas in three-letter groups. I won't spoil the fun of playing the game by telling you what the telegram must say, although close analysis of the program will give you the answer.

The three subroutines between Lines 2970 and 3510 read numeric and alphabetic data into the proper variables. The subroutine at Lines 3530–3570 checks for a yes or no answer to an INPUT statement. The telegraph and warning beeper subroutines make simple sounds, the "Ready to go?" subroutine halts the program until a key is pressed, while the pause subroutine creates a short pause in program execution.

The simulated race can end in one of two ways: You make it to Paris or you don't. If you make it, the routine at Lines 3740 to 3870 tells you how long it took and where you placed in the race. If you don't make it, a consoling message is printed, and you are given a chance to try again.

PROGRAM VARIABLES

A Answer of user (0 = Yes, 1 = No)
A\$ Answer to string input query, user input
AP Telegram-politeness indicator
AT\$ 3-letter group from telegram
AS Telegram-urgency indicator
C(n) Road conditions by location, index of C\$(n), n = 1 - 20
C\$(n) Road-conditions descriptor, n = 1 - 20
D Distance of segment
DA Distance, cumulative within segment
DC Distance, cumulative
DD Distance, daily
DX(n) Distance of segment, n = 1 - 20
F Mechanical failure, index for FA\$, FB\$, FC\$, FL, FT
FA\$(n) Mechanical-failure descriptor, n = 1 - 18
FB\$(n) Mechanical-failure fix descriptor, n = 1 - 18
FC\$(n) Mechanical-failure 2nd fix descriptor, n = 1 - 18
FD\$ Mechanical failure, hours or days descriptor
FL(1,n) Mechanical-failure cost of 1st fix, n = 1 - 18
FL(2,n) Mechanical-failure cost of 2nd fix, n = 1 - 18
FQ Mechanical-failure, how to fix, user input
FT(1,n) Mechanical-failure time for 1st fix, n = 1 - 18
FT(2,n) Mechanical-failure time for 2nd fix, n = 1 - 18
FU Mechanical failure, fix time
FX Mechanical-failure indicator
GF Gasoline base price per gallon
GG Gasoline reserve
GM Gasoline used by day
GP Gasoline price
HC Hours per day, cumulative
HP Hours per day to drive, user input
I Temporary index
J Journey-segment number
JV Ocean-voyage indicator
K Temporary indicator and index
LA\$(n) City, n = 1 - 20
LB\$(n) State or country, n = 1 - 20
MD Day of month

MO\$ Month
 PB Probability of mechanical breakdown
 PF Probability of fatigue problem
 PW Probable speed factor due to weather and road conditions
 RN Random variable for random number
 RQ Random variable for hours
 SP Speed, user input
 T Time-into-race segment

 TD Time, cumulative
 TE Time of 1908 winning car to complete segment
 TL Time, cumulative of 1908 race leader
 TT Temporary date indicator
 TX(n) Time of 1908 winning car to complete segment, n = 1 - 20
 TZ Time delay
 W Weather on the trip, index of w\$(n)
 W\$(n) Weather descriptor, n = 1 - 6
 WX(n) Weather, probable at each location, n = 1 - 20
 X Temporary variable
 X\$ Temporary string variable
 Z Cash balance of racing team
 ZN Cash expenditure during race
 Note: all variables use the following measurement units:

D	Distance	Miles
G	Gasoline	Gallons
H	Time	Hours
P	Probabilities	0 to 1
T	Time	Days
Z	Money	Dollars and cents

PROGRAM LISTINGS

[Download AUTORACE.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF
110 LOCATE 10, 23 : PRINT "The Longest Automobile Race, 1908" : PRINT
120 PRINT : PRINT TAB(29) "(c) David H. Ahl, 1986" : LOCATE 23, 21
130 '
140 'Initialization
150 DIM LA$(20), LB$(20), TX(20), C(20), WX(20), DX(20)
160 DIM FA$(20), FB$(20), FC$(20), FT(2, 20), FL(2, 20)
170 Z = 1000 : GF = .25 : 'Starting cash and gas price
180 GOSUB 2970 : 'Initialize text variables
190 GOSUB 3060 : 'Initialize location data
200 GOSUB 3310 : 'Initialize mechanical breakdown data
210 PRINT "Press any key when you're ready to go" : RN = -32768!
220 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND
230 WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN : GOSUB 4010
240 '
250 'Main program
260 J = J + 1 : T = 0 : GOSUB 3710 : GOSUB 3630 : 'New location, road data, etc.
270 DA = 0 : W = WX(J) : D = DX(J) : TE = TX(J) : 'Set variables for new location
280 GOSUB 940 : 'Print the date
290 PRINT "You are at " LA$(J) ", " LB$(J) "."
300 PRINT "You currently have"; : PRINT USING "$$####.##";Z : IF J = 1 THEN 420
310 IF FX = 0 THEN 340 : 'Any unfixed mechanical parts?
320 PRINT "A sympathetic garage owner will fix the " FA$(FX) " here." : FX = 0
330 TZ = INT(1 + 3 * RND(1)) : PRINT "It will take" TZ "day(s)." : GOSUB 2590
340 IF J > 7 AND J < 11 THEN GOSUB 2290 : 'Ocean voyage?
350 IF JV = 1 THEN JV = 0 : TL = TL + TE : GOTO 260 : 'End of ocean voyage?
360 PRINT : PRINT "You have driven" INT(DC) "miles in" TD "days."
370 IF J = 20 THEN 3740 : 'Finished the race?
380 IF TD < TL THEN 400 ELSE IF TD = TL THEN 410
390 PRINT "The race leader passed this point" TD - TL "day(s) ago." : GOTO 420
400 PRINT "You are the race leader and are" TL - TD "day(s) ahead." : GOTO 420
410 PRINT "You and the Italian Zusto are running even with each other."
420 TL = TL + TE : 'Elapsed time of race leader
430 IF J = 7 OR J = 12 THEN GOSUB 2290 : 'Ocean voyage after land segment?
440 IF JV = 1 THEN JV = 0 : GOTO 260 : 'End of ocean voyage?

```

```

450 PRINT "Roads to the west of here are " C$(C(J)) "."
460 PRINT "The weather forecast is " W$(W) "."
470 PRINT "You set a goal of making" D "miles in the next" TE - 2 "days."
480 GOSUB 640 : 'Buy gas and oil
490 GOSUB 730 : 'Get desired speed
500 GOSUB 820 : 'Get desired driving hours per day
510 '
520 'Go through this race segment day by day
530 TZ = 1 : GOSUB 2590 : 'Increment through each day of travel
540 GOSUB 1120 : 'Weather subroutine
550 GOSUB 1640 : 'Mechanical breakdown subroutine
560 GOSUB 1930 : 'Accident subroutine
570 DD = SP * HP * PW : DA = DA + DD : DC = DC + DD : 'Daily and cumulative distances
580 GM = .07 * DD * (.8 + .4 * RND(1)) : 'Gas used today
590 IF GM < GG THEN GG = GG - GM : GOTO 620 : 'Subtract gas used from supply
600 BEEP : BEEP : BEEP : GOSUB 940 : PRINT "You ran out of gas on the road."
610 GF = .33 : GOSUB 640 : GG = GG - GM : 'Buy gas
620 IF DA >= D THEN 260 ELSE 520 : 'Complete a travel segment?
630 '
640 'Fuel and oil subroutine
650 GP = GF * (.7 + .6 * RND(1)) : PRINT "Gas costs" INT(100 * GP) "cents per gallon here."
660 INPUT "How many gallons do you want for the segment ahead"; GG
670 GF = .25 : ZN = GG * GP : PRINT "That will cost"; : PRINT USING "$$###.##";ZN
680 GOSUB 2660 : 'Do you have enough money?
690 IF A = 0 THEN RETURN : 'Enough money now?
700 IF Z < 2 THEN PRINT "Your car won't run on fumes. It's all over." : GOTO 3910
710 GG = INT(Z / GP) : PRINT "Sorry, you could only get" GG "gallons." : RETURN
720 '
730 'Input desired speed subroutine
740 INPUT "How fast (mph) do you want to drive";SP
750 IF SP > 54 THEN PRINT "Top speed of your car is only 54 mph." : GOTO 740
760 IF SP < 8 THEN PRINT "At that rate, you'll never get there." : GOTO 740
770 IF W < 3 AND SP > 30 THEN 780 ELSE 790
780 PRINT "That's too fast for these weather and road conditions." : GOTO 740
790 PB = SP * SP / 7000 : 'Probability of mechanical breakdown is related to speed
800 RETURN
810 '
820 'Input desired driving hours subroutine
830 K = 0 : 'Counter for pushing too hard
840 INPUT "How many hours do you want to drive each day"; HP
850 IF K = 1 THEN 910 : 'Did we ask about pushing too hard already?
860 IF HP > 8 THEN PRINT "That's too much for both you and your car." : GOTO 840
870 IF HP < 2 THEN PRINT "No one is that lazy!" : GOTO 840
880 HC = HC + HP : IF J > 2 AND HC / J > 7.55 THEN 890 ELSE 910
890 PRINT "You've been pushing yourself and your crew pretty hard."
900 PRINT "You should probably back off a bit." : K = 1 : GOTO 840
910 PF = HP ^ 3 / 1000 - .15 : IF PF < .01 THEN PF = .01 : 'Probability of fatigue problem
920 RETURN
930 '
940 'Date subroutine
950 IF TT = TD THEN RETURN : 'Printed this date already?
960 IF TD < 19 THEN 990 ELSE IF TD < 50 THEN 1000 ELSE IF TD < 80 THEN 1010
970 IF TD < 111 THEN 1020 ELSE IF TD < 141 THEN 1030 ELSE IF TD < 172 THEN 1040
980 IF TD < 203 THEN 1050 ELSE GOSUB 3630 : GOTO 1070
990 MO$ = "February" : MD = TD + 11 : GOTO 1060
1000 MO$ = "March" : MD = TD - 18 : GOTO 1060
1010 MO$ = "April" : MD = TD - 49 : GOTO 1060
1020 MO$ = "May" : MD = TD - 79 : GOTO 1060
1030 MO$ = "June" : MD = TD - 110 : GOTO 1060
1040 MO$ = "July" : MD = TD - 140 : GOTO 1060
1050 MO$ = "August" : MD = TD - 171
1060 PRINT : PRINT "Date: " MO$; MD ", 1908" : TT = TD : RETURN
1070 PRINT : PRINT "It's September 1 and the winning car crossed the finish"
1080 PRINT "line in Paris over a month ago. Your factory refuses to give"
1090 PRINT "you any more money to continue. Better luck next time."
1100 GOTO 3890
1110 '
1120 'Weather subroutine
1130 ON W GOTO 1150, 1230, 1290, 1350, 1350, 1490
1140 '
1150 'Heavy snow and blizzard conditions

```



```

1160 RN = RND(1) : IF RN < .33 THEN 1170 ELSE IF RN > .83 THEN 1190 ELSE 1210
1170 GOSUB 960 : PW = .03 + .08 * RND(1) : 'Speed factor in blizzard
1180 PRINT "Blizzard conditions. Tough going today." : GOSUB 3710 : RETURN
1190 GOSUB 960 : PW = .05 + .1 * RND(1) : 'Speed factor in very heavy snow
1200 PRINT "You're stuck in a huge snow drift." : GOSUB 1530 : RETURN
1210 PW = .14 + .17 * RND(1) : RETURN : 'Speed factor on heavy snow day
1220 '
1230 'Snow conditions
1240 IF RND(1) < .1 THEN 1260 : '10% chance of getting stuck in snow
1250 PW = .3 + .4 * RND(1) : RETURN : 'Speed factor on normal snowy day
1260 PW = .15 + .1 * RND(1) : GOSUB 960 : PRINT "You have skidded into a ditch."
1270 GOSUB 1530 : RETURN : 'Hire farmer to pull you out of ditch?
1280 '
1290 'Rainy weather
1300 IF RND(1) < .2 THEN GOSUB 960 : GOTO 1320 : 'Bogged down in mud?
1310 PW = .35 + .4 * RND(1) : RETURN : 'Speed factor on normal rainy day
1320 PW = .02 + .04 * RND(1) : PRINT "You are totally bogged down in the mud."
1330 GOSUB 1530 : RETURN : 'Hire farmer to pull you out of mud?
1340 '
1350 'Cloudy and mixed weather
1360 RN = RND(1) : IF RN > .08 THEN 1470 ELSE GOSUB 960 : 'Normal cloudy day?
1370 IF RN < .01 THEN PRINT "An unexpected downpour!" : GOTO 1320
1380 PRINT "River ahead with no bridge. Some locals tell you there is a bridge"
1390 PRINT "'some distance' north. They also offer to take you across by boat"
1400 ZN = 3 + 2 * INT(3 * RND(1)) : PRINT "for $" ZN; : INPUT ". Want to go by boat";A$
1410 GOSUB 3530 : IF A = 1 THEN 1450 : 'Not willing to pay?
1420 GOSUB 2660 : 'Go to pay the bill routine
1430 IF A = 1 THEN 1450 : 'Still not enough money?
1440 PRINT "They got you across in" 2 + INT(3 * RND(1)) "hours." : PW = .3 : RETURN
1450 TZ = INT(1 + 2 * RND(1)) : PRINT "It took" TZ "day(s) for you to drive north ";
1460 PRINT "and find the bridge." : GOSUB 2590 : RETURN
1470 PW = .4 + .4 * RND(1) : RETURN : 'Normal cloudy day
1480 '
1490 'Clear and sunny
1500 IF RND(1) < .025 THEN GOSUB 960 : GOTO 1380 : 'River with no bridge
1510 PW = .45 + .5 * RND(1) : RETURN : 'Normal sunny day
1520 '
1530 'Pull you out of ditch subroutine
1540 ZN = 5 * INT(1 + 4 * RND(1)) : PRINT "A farmer offers to pull you out for $" ZN
1550 INPUT "Do you want to pay him to pull you out";A$ : GOSUB 3530
1560 IF A = 1 THEN 1610 : 'Not willing to pay?
1570 GOSUB 2660 : 'If not enough money try to get some
1580 IF A = 1 THEN 1610 : 'Still not enough money?
1590 RQ = INT(1.5 + 5 * RND(1)) : PRINT "It took" RQ "hours for him to pull you out."
1600 IF RQ < 5 THEN RETURN ELSE TZ = 1 : GOSUB 2590 : PW = PW * 1.5 : RETURN
1610 TZ = INT(1 + 1.3 * RND(1)) : PRINT "It took" TZ "day(s) for you and your mechanic"
1620 PRINT "to pull the car out by yourselves." : GOSUB 2590 : PW = PW * 1.5 : RETURN
1630 '
1640 'Mechanical breakdowns subroutine
1650 IF RND(1) > PB THEN RETURN : 'If no mechanical breakdown then return
1660 F = INT(1 + 15 * RND(1)) : IF F > 13 THEN F = INT(14 + 5 * RND(1)) : 'What malfunction
1670 BEEP : BEEP : BEEP : GOSUB 940 : 'Sound warning beeper and print date
1680 PRINT "Uh oh. You have a problem. It's a " F$(F) "."
1690 PRINT "Here's what you can do about the problem:"
1700 PRINT TAB(7) "(1) Try to keep going with it"
1710 PRINT TAB(7) "(2) " F$(F) ", cost $" FL(1, F)
1720 IF F$(F) = "" THEN 1740 : 'Only one way to fix it?
1730 PRINT TAB(7) "(3) " F$(F) ", cost $" FL(2, F)
1740 INPUT "Which would you like to do";FQ
1750 IF FQ = 1 THEN 1840 : 'Decided to do nothing?
1760 IF FQ = 2 OR FQ = 3 THEN 1770 ELSE PRINT "Please enter a number." : GOTO 1740
1770 FQ = FQ - 1 : FU = FT(FQ, F) : IF FU < 8 THEN FD$ = "hours" ELSE 1790
1780 IF FU < 5 THEN 1810 ELSE TZ = 1 : GOTO 1800 : 'Lose a day for repairs?
1790 FU = FT(FQ, F) / 8 : TZ = FU : IF FU = 1 THEN FD$ = "day" ELSE FD$ = "days"
1800 PW = PW * 1.5 : GOSUB 2590 : 'Allow for 1/2 day driven & increase day counters
1810 ZN = FL(FQ, F) : PRINT "Repairs will take" FU;FD$ " and will cost $" ZN
1820 GOSUB 2660 : 'Pay the repair bill
1830 IF A = 0 THEN RETURN : 'Enough money?
1840 PRINT "You try to nurse the car along to the next major city."
1850 IF FX = 0 THEN 1870 : 'Any unfixed malfunction?
1860 PRINT "But with the other problem you just can't make it and" : GOTO 1890

```

```

1870 IF RND(1) > .4 THEN 1900 : '60% chance that you can nurse it along
1880 GOSUB 3710 : PRINT : PRINT "Unfortunately, it just won't make it and"
1890 PRINT "reluctantly you admit defeat." : GOTO 3890
1900 PRINT "It looks like you'll make but at a drastically reduced speed."
1910 PW = PW * .5 : FX = F : RETURN : 'Cut speed factor in half, note unfixed item
1920 '
1930 'Accidents and special situations subroutine
1940 IF RND(1) > PF THEN 2100 : 'If driving long hours is not a problem, go on
1950 BEEP : BEEP : BEEP : BEEP : GOSUB 940 : PRINT "You dozed off and your car has run ";
1960 ON INT(1 + 4 * RND(1)) GOTO 1970, 1980, 1990, 2000, 2000
1970 PRINT "into a tree." : TZ = 2 : ZN = 24 : GOTO 2010
1980 PRINT "off the road." : TZ = 1 : ZN = 12 : GOTO 2010
1990 PRINT "into a gaping hole." : TZ = 1 : ZN = 18 : GOTO 2010
2000 PRINT "into a farmer's wagon." : TZ = 2 : ZN = 25
2010 PRINT "You can try to fix it or get a tow to the next village for $15."
2020 INPUT "Want to try to bang out the damage on the spot";A$ : GOSUB 3530
2030 IF A = 0 THEN 2070 ELSE PRINT "The tow costs $15 and the repairs cost $" ZN
2040 ZN = ZN + 15 : GOSUB 2660 : 'Pay the bills
2050 IF A = 0 THEN 2100 : 'Enough money now?
2060 PRINT "The locals impound your car for your unpaid debt." : GOTO 3890
2070 PRINT "You finally manage to do it but it takes" TZ "day(s)."
2080 PW = PW * 1.5 : GOSUB 2590 : 'Allow for 1/2 day driven & increase day counters
2090 '
2100 'Drive on railraod ties routine
2110 IF J< >2 AND J< >5 AND J< >13 AND J< >14 THEN 2190 : 'In area with railroads?
2120 IF RND(1) > .4 THEN 2190 ELSE GOSUB 940 : '40% chance to drive on rwy tracks
2130 PRINT "In this area of terrible roads, you can save some time by driving"
2140 PRINT "on the railraod tracks. However, it is murder on your wheels,"
2150 PRINT "tires, and whole car." : INPUT "Want to drive on the tracks";A$
2160 GOSUB 3530 : IF A = 1 THEN RETURN
2170 PW = PW * 1.7 : PB = PB * 1.25 : RETURN : 'Increase speed and chance of trouble
2180 '
2190 'No grease routine
2200 IF J< >15 AND J< >16 THEN RETURN : 'In central Russia?
2210 IF RND(1) > .2 THEN RETURN : '20% chance of no grease
2220 BEEP : BEEP : BEEP : GOSUB 940 : PRINT "Your differential is dry and there is"
2230 PRINT "no grease available here. However, you can get Vaseline."
2240 INPUT "Want to use it in place of grease";A$ : GOSUB 3530
2250 IF A = 0 THEN PRINT "Okay, you buy 20 jars for $4." : Z = Z - 4 : RETURN
2260 PRINT "The gears sound horrible. You'll have to cut your speed in half."
2270 PW = PW * .5 : RETURN
2280 '
2290 'Ocean voyage subroutine
2300 JV = 1 : TZ = INT(1 + 3.5 * RND(1)) : 'Length of time in port
2310 IF J = 12 THEN 2510 ELSE IF J = 10 THEN 2480
2320 IF J = 9 THEN 2420 ELSE IF J = 8 THEN 2390
2330 PRINT "You're stuck in port for" TZ + 1 "days before you can get a steamer"
2340 PRINT "for Seattle. You use the time to get new countershaft"
2350 PRINT "housings, springs, wheels, drive chains, and tires."
2360 IF Z > 300 THEN PRINT "The cost of these items is $164." : Z = Z - 164 : GOTO 2380
2370 PRINT "These were all furnished by the local Thomas Flyer dealer."
2380 TZ = TZ + 1 : TD = TD + 3 : GOSUB 2590 : GOSUB 3660 : RETURN
2390 PRINT "It took 3 days on the steamer. The next steamer for Valdez"
2400 PRINT "Leaves in" TZ "days. Nothing to do but wait." : GOSUB 2590
2410 GOSUB 3660 : TD = TD + 7 : RETURN
2420 PRINT "The steamer made many stops up the coast and it took 7 days."
2430 PRINT "It is apparent that the race organizers have never been in Alaska"
2440 PRINT "and have no idea that it is impossible to drive on the snow and"
2450 PRINT "ice at all, much less across the Bering Strait to Russia. You'll"
2460 PRINT "have to return to Seattle. Next steamer goes in" TZ "days."
2470 GOSUB 2590 : GOSUB 3660 : TD = TD + 7 : RETURN
2480 PRINT "It took 7 days to get back to Seattle. Now you have a" TZ "day"
2490 PRINT "wait before you can get a freighter for Japan."
2500 GOSUB 2590 : GOSUB 3660 : TD = TD + 21 : RETURN
2510 PRINT "The freighter across the Pacific takes a leisurely 21 days making"
2520 PRINT "stops at Hawaii, Guam, and the Philippines. Also the Chinese"
2530 PRINT "crewmen made sandals out of your leather fenders and mud flaps."
2540 PRINT "You can't replace them in Japan, but you can at Vladivostock,"
2550 PRINT "Russia. There you'll have to spend several days arranging for"
2560 PRINT "fuel also. But hurry now. A steamer to Russia leaves tonight."
2570 GOSUB 3660 : TD = TD + 7 : RETURN

```

```

2580 '
2590 'Time delay and hotel bills routine
2600 T = T + TZ : TD = TD + TZ : 'Increment time counters
2610 ZN = 10 * TZ : GOSUB 2660 : 'Pay for hotel and meals
2620 IF A = 0 THEN RETURN : 'Did we have enough money?
2630 PRINT : PRINT "You don't even have enough money to pay for meals."
2640 PRINT "That's the end of the road for you." : PRINT : GOTO 3890
2650 '
2660 'Pay the bills routine
2670 IF Z < ZN THEN GOSUB 2710 ELSE 2690 : 'Enough money to pay bills?
2680 IF Z < ZN THEN A = 1 : RETURN : 'Couldn't get money from home
2690 Z = Z - ZN : A = 0 : RETURN : 'Subtract money from kitty
2700 '
2710 'Need more money subroutine
2720 ZB = ZB + 1 : IF ZB < 3 THEN ZW = 1000 ELSE ZW = 500 : 'Money request and amount
2730 PRINT : PRINT "You don't have enough money to continue. Your only hope is"
2740 PRINT "to send a telegram back to Mr. Thomas at the factory and ask"
2750 PRINT "for more money. Remember, telegrams in 1908 used all capital"
2760 PRINT "letters, had no commas, and were short."
2770 INPUT "What is your message";A$: PRINT "Sending telegram now..."
2780 GOSUB 3590 : IF ZB > 3 THEN 2940
2790 AS = 0 : AP = 0 : L = LEN(A$) : IF L < 12 THEN 2920
2800 FOR I = 1 TO L - 2 : AT$ = MID$(A$, I, 3) : 'Look at 3-letter groups in telegram
2810 IF AT$ = "PLE" OR AT$ = "BEG" OR AT$ = "SOR" OR AT$ = "IMP" THEN AP = 1
2820 IF AT$ = "SOO" OR AT$ = "QUI" OR AT$ = "EAR" OR AT$ = "FAS" OR AT$ = "HUR" THEN AS = 1
2830 IF AT$ = "IMM" OR AT$ = "ONC" OR AT$ = "URG" THEN AS = 1
2840 NEXT I : IF AP = 0 THEN 2880 ELSE IF AS = 0 THEN 2860
2850 PRINT "Mr. Thomas wired back $ "ZW" and said 'GOOD LUCK!" : Z = Z + ZW : RETURN
2860 PRINT "Mr. Thomas didn't know you needed the money right away and waited"
2870 PRINT "3 days before wiring back $" ZW : Z = Z + ZW : TZ = 3 : GOSUB 2590 : RETURN
2880 IF AS = 0 THEN 2910
2890 PRINT "Mr. Thomas wired back, 'YOU COULD AT LEAST BE POLITE,' but did"
2900 ZW = ZW / 2 : PRINT "include a draft for $" ZW : Z = Z + ZW : RETURN
2910 PRINT "Mr. Thomas was offended by your telegram and refused to" : GOTO 2930
2920 PRINT "Your message was short all right. Too short. Mr. Thomas didn't"
2930 PRINT "send any money. Sorry." : RETURN
2940 PRINT "Mr. Thomas wires back: I AM FED UP WITH THIS ADVENTURE STOP"
2950 PRINT "YOU WILL GET NO MORE MONEY FROM ME STOP" : RETURN
2960 '
2970 'Subroutine to put verbal data into constants
2980 FOR I = 1 TO 6 : READ C$(I) : NEXT I
2990 FOR I = 1 TO 6 : READ W$(I) : NEXT I
3000 DATA "hard packed gravel", "muddy ruts", "slightly improved wagon tracks"
3010 DATA "built for narrow carts", "practically non-existent", "horrible"
3020 DATA "blizzard conditions", "snow and sleet", "rain"
3030 DATA "cloudy with a chance of rain", "mixed", "sunny and dry"
3040 RETURN
3050 '
3060 'Location, expected weather, road conditions, need to use rail,
3070 ' expected days to next location, distance to next location
3080 FOR I = 1 TO 20 : READ X, LA$(I), LB$(I), WX(I), C(I), TX(I), DX(I) : NEXT I
3090 RETURN
3100 DATA 1, "New York", "New York", 2, 1, 8, 897
3110 DATA 2, "Kendallville", "Indiana", 1, 1, 6, 166
3120 DATA 3, "Chicago", "Illinois", 3, 2, 7, 634
3130 DATA 4, "Omaha", "Nebraska", 6, 3, 4, 482
3140 DATA 5, "Laramie", "Wyoming", 2, 3, 7, 467
3150 DATA 6, "Ogden", "Utah", 6, 1, 8, 1237
3160 DATA 7, "San Francisco", "California", 5, 7, 8, 0
3170 DATA 8, "Seattle", "Washington", 5, 7, 8, 0
3180 DATA 9, "Valdez", "Alaska", 5, 7, 8, 0
3190 DATA 10, "Seattle", "Washington", 5, 7, 25, 0
3200 DATA 11, "Kobe", "Japan", 4, 4, 4, 350
3210 DATA 12, "Tsuruga", "Japan", 4, 7, 7, 0
3220 DATA 13, "Vladivostock", "Russia", 3, 5, 15, 558
3230 DATA 14, "Tsitsihar", "Manchuria", 5, 6, 10, 659
3240 DATA 15, "Chita", "Russia", 3, 3, 8, 1116
3250 DATA 16, "Kansk", "Russia", 4, 3, 6, 1075
3260 DATA 17, "Omsk", "Russia", 5, 1, 7, 820
3270 DATA 18, "Perm", "Russia", 3, 2, 14, 1090
3280 DATA 19, "St. Petersburg", "Russia", 3, 1, 8, 1575

```

```

3290 DATA 20, "Paris", "France", 0, 0, 0, 0
3300 '
3310 'Mechanical breakdown descriptions, time to fix, cost
3320 FOR I = 1 TO 18
3330 READ X, FA$(I), FB$(I), FC$(I), FT(1, I), FL(1, I), FT(2, I), FL(2, I) : NEXT I : RETURN
3340 DATA 1, "tire blowout", "Patch the hole", "Replace the tire", 2, 1, 2, 7
3350 DATA 2, "skipping cylinder", "New spark plugs", "Grind cylinder", 1, 2, 8, 2
3360 DATA 3, "rough running engine", "Do a tune up", "", 4, 5, 0, 0
3370 DATA 4, "binding axle bearing", "Regrind bearing", "Get a new one", 8, 2, 4, 8
3380 DATA 5, "cracked spring", "New spring", "Weld angle iron to it", 8, 26, 8, 4
3390 DATA 6, "cracked wheel", "New wheel", "Weld brace on it", 2, 42, 8, 4
3400 DATA 7, "slipping clutch", "Adjust clutch", "New clutch plate", 4, 4, 8, 54
3410 DATA 8, "stripped gear", "Weld teeth back on", "New gear", 16, 6, 8, 24
3420 DATA 9, "radiator leak", "Weld a patch on it", "", 4, 2, 0, 0
3430 DATA 10, "brakes failure", "Replace the linings", "", 8, 7, 0, 0
3440 DATA 11, "crack in the countershaft housing", "A new housing", "", 24, 40, 0, 0
3450 DATA 12, "broken drive pinion", "Weld teeth back on", "New pinion", 16, 6, 8, 18
3460 DATA 13, "broken rear axle", "Get a new axle", "", 16, 68, 0, 0
3470 DATA 14, "cracked transmission housing", "New one from factory", "", 24, 60, 0, 0
3480 DATA 15, "broken motor support", "Make a new one of scrap iron", "", 16, 16, 0, 0
3490 DATA 16, "worn down clutch shaft", "A new clutch shaft", "", 8, 28, 0, 0
3500 DATA 17, "cracked frame", "Weld on braces of angle iron", "", 24, 26, 0, 0
3510 DATA 18, "total transmission failure", "A new one from factory", "", 40, 225, 0, 0
3520 '
3530 'Check for yes or no answer
3540 IF A$ = "" OR A$ = "Y" OR A$ = "y" THEN A = 0 : RETURN
3550 IF A$ = "N" OR A$ = "n" THEN A = 1 : RETURN
3560 PRINT "Don't understand your answer of ";A$; "."
3570 INPUT "Please enter Y for 'yes' or N for 'no.'";A$ : GOTO 3530
3580 '
3590 'Telegraph routine
3600 FOR I = 1 TO 4 : X = 1 + 3 * RND(1) : FOR K = 1 TO X : BEEP : NEXT K
3610 FOR K = 1 TO 500 : NEXT K : NEXT I : RETURN
3620 '
3630 'Warning beeper routine
3640 PRINT : FOR I = 1 TO 3 : BEEP : BEEP : FOR K = 1 TO 500 : NEXT K : NEXT I : RETURN
3650 '
3660 'Ready to go? routine
3670 PRINT : PRINT "Press any key when you're ready to go aboard."
3680 WHILE LEN(INKEY$) = 0 : WEND
3690 RETURN
3700 '
3710 'Pause routine
3720 FOR I = 1 TO 500 : NEXT I : RETURN
3730 '
3740 'Finished the race in Paris!
3750 FOR K = 1 TO 3 : GOSUB 3720 : BEEP : BEEP : BEEP : NEXT K : CLS : X = 0
3760 FOR I = 1 TO 30 : FOR K = 1 TO 100 : NEXT K : LOCATE 10, 30 : PRINT X$ : BEEP
3770 IF X = 0 THEN X$ = "CONGRATULATIONS!" : X = 1 : GOTO 3790
3780 X$ = "" : X = 0 : 'X$ = 17 spaces to erase congratulations
3790 NEXT I : PRINT : PRINT
3800 IF TD < TL THEN 3830 ELSE IF TD = TL THEN 3850
3810 PRINT "You made it to Paris! The German Protos beat you by"
3820 PRINT TD - TL "days but just to finish is a great honor!" : GOTO 3860
3830 PRINT "You reached Paris first! The next car is" TL - TD "days behind."
3840 GOTO 3860
3850 PRINT "You reached Paris in a dead tie with the French Motobloc!"
3860 PRINT : PRINT "You reached Paris in" TD "days. In 1908, the Thomas Flyer"
3870 PRINT "won the race reaching Paris on July 30 after 169 days." : GOTO 3970
3880 '
3890 'End of race and summary statistics
3900 PRINT : GOSUB 3630 : PRINT
3910 PRINT "Sorry you were unsuccessful. Only three of the"
3920 PRINT "cars in the 1908 race ever finished." : PRINT
3930 PRINT "In the" TD "days since the start of the race on February 12, 1908,"
3940 PRINT "you covered" INT(DC) "miles. You almost made it to"LA$(J + 1) ", ";
3950 PRINT LB$(J + 1) ". " : PRINT "Not bad, but you can do better." : PRINT
3960 '
3970 PRINT : INPUT "Would you like to try again (Y or N)";A$ : GOSUB 3530
3980 IF A = 0 THEN PRINT "Okay. Good Luck!" : GOSUB 3720 : CLS : RUN
3990 PRINT "Okay. So long for now." : GOSUB 3720 : KEY ON : CLS : END

```

```
4000 '  
4010 'Subroutine to print the instructions  
4020 CLS : PRINT TAB(18) "The Longest Automobile Race, 1908" : 1908 : PRINT  
4030 PRINT " In this program, you are the captain of the Thomas Flyer team."  
4040 PRINT "It is your job to get the car from New York to Paris—east to west—  
4050 PRINT "as quickly as possible. The race starts on Febraury 12, 1908."  
4060 PRINT " You must overcome many problems : bad weather, accidents,"  
4070 PRINT "mechanical breakdowns, fatigue, and a lack of gas stations."  
4080 PRINT " For each leg of the trip, buy as much gas as you need, but no"  
4090 PRINT "more. You car gets approximately 14 mph, although this will vary."  
4100 PRINT "You will carry what fuel you can and ship the rest ahead by rail to"  
4110 PRINT "locations along your route to be held for you (called 'spotting')."  
4120 PRINT " Your car has a top speed of 54 mph. However, the probability"  
4130 PRINT "of a breakdown increases substantially at speeds over 35 mph. Like-  
4140 PRINT "wise, driving more than six hours per day increases your chance of"  
4150 PRINT "having an accident. But don't forget, this IS a race."  
4160 PRINT " If you get stuck, you can pay someone to pull you out (costs"  
4170 PRINT "money) or try to get out on your own (costs time)."  
4180 PRINT " You can choose to repair a mechanical problem on the spot or"  
4190 PRINT "wait until the next large town to get it fixed. Either choice has"  
4200 PRINT "associated risks."  
4210 PRINT " If and when you run out of money, you can wire Mr. Thomas for"  
4220 PRINT "more, but your request must be carefully and politely worded Also,"  
4230 PRINT "your telegram must be in all UPPER CASE letters."  
4240 PRINT TAB(21) "Press any key to continue." ;  
4250 WHILE LEN(INKEY$) = 0 : WEND : CLS : RETURN
```

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



THE GAME

The Orient Express places you aboard that famous train bound from London, England to Constantinople in February 1923. As a secret agent your assignment is to make contact with an arms dealer who has tired of his life of crime, and arrest the killer of one of his former associates.

Some of the passengers on the train (all of whom are historical characters who actually rode the Express in their day) have information that can help you complete your assignment. As you work your way through the mystery, you participate in many of the things—delicious meals, stops at stations along the way, and occasional delays due to snow, derailment, and bandits—that made the Orient Express an unforgettable experience for those who rode it.

Solving the puzzle is not easy. You will probably want to make notes about meaningful clues and take some time to think—hours perhaps—before talking to the Turkish police (who will help you) at Uzunkopru. Bear the following in mind as you play the game:

- You must identify and protect the sender of the note and identify and arrest the killer of Baron Wunster.
- Five notorious arms dealers, all of different nationalities, are currently operating in Europe under an uneasy truce. Each of them deals in a different type of weapon, and all are known to have different tastes and habits.
- Not all of the passengers with whom you speak have useful information. You cannot talk to passengers during sleeping segments of the trip.
- Unless you have extraordinary deductive powers, it will probably take you at least an hour of play to solve the mystery. (Of course, you can cheat and do some sleuthing in the program listing, but that will probably

take you almost as long and be a lot less fun.)

So grab your notebook and your ticket, and start asking questions. Remember, a man's life is at stake!

THE STORY

What image comes to mind at the mention of the Orient Express? Adventure? Luxury? Intrigue? Reliability? Awe? The famous train was all of these things and much more. Very few people ever traveled on the Orient Express, even at its peak, because the cost was prohibitive for all but the super rich. However, no form of public transportation has fascinated more authors, journalists, and filmmakers than the Orient Express. Thus, it has assumed a dimension almost larger than life. D.H. Lawrence had Lady Chatterley and her lover travel on the Orient Express. Agatha Christie, who was, in fact, a regular traveler on the Orient Express, used it as the setting for one of her most compelling Hercule Poirot stories, *Murder on the Orient Express*. Graham Greene's *Stambol Train* is about the Orient Express. Eric Ambler also used it as the setting for *The Mask of Dimitrios*, and one of the most accurate descriptions of the postwar train is found in the final chapters of Ian Fleming's *From Russia with Love*. And there are scores of others. How did this magnificent train come into existence and how did it achieve such fame and notoriety? In the 30 or so years following 1825, when the first steam railroad was put into operation in England, every country in Europe began building its own railroad system. Political conditions were unsettled, and the threat of war hung heavy in the air. Hence railway builders would deliberately use track widths and rolling stock incompatible with those of their neighbors to thwart the movement of invaders by rail. Transcontinental travelers thus had to change trains at every border, and frequently used a ship or stagecoach to make connections.

Moreover, railroad coaches had evolved from horse-drawn carriages and were quite uncomfortable. Trains had no lavatories, no restaurant cars, no connections between coaches, no corridors, no lighting, hard seats, and little heat. Although progress was being made in Europe, it was in America that the real advances in rail travel were taking place. There, consistent with the democratic principles on which the country had been founded, railroad cars had open seating in contrast with the tiny boxed-in compartments found on European trains. Distances in the U.S. were longer, so provision had to be made for eating and other necessities. The three most important advances in American railroading sprang from necessity. First, much track in America had been poorly laid, so derailments were common. To overcome this problem, the fixed wheel-and-axle design was discarded, and railroad cars were equipped with bogies, swiveling four-wheel trucks with independent springs. Second, American

trains were much longer and heavier than European trains, and the locomotive brakes required a very long distance to stop a train. So, for greater efficiency, George Westinghouse developed a system of compressed air brakes. The third advance was George Pullman's development of a combined parlor/sleeping car. Georges Nagelmackers, son of a wealthy Belgian banker, journeyed to the U.S. in 1869 and was very impressed by American railroads and particularly by Pullman's luxurious cars. By the time he returned to Europe in 1870, Nagelmackers was obsessed with the idea of establishing luxury, long-distance, through rail service throughout Europe.

His indulgent father was impressed with his son's enthusiasm and proposed to King Leopold II of Belgium that he would forgive the King's substantial overdue bank loans if the King would agree to head the list of subscribers to the venture. The King, seeing a way out of a difficult financial situation, agreed, and the new company of Nagelmackers et Cie. gained an endorsement. Soon their stock sales promotion had every social climber in Belgium clamoring to get aboard.

The new company's first train was to run from Paris to Berlin by way of Belgium. Five cars, ordered from a coach-building company in Vienna, arrived in early July to begin service later in the month. Then, on July 19, 1870, France declared war on Prussia. Discouraged, but still determined, Nagelmackers negotiated a route from Ostend, Belgium, through France, and down to Brindisi. The route was very successful—for one year. Then, on September 17, 1871, the French opened a tunnel through the Alps under Mount Cenis. Rights were reserved for French-owned rail-roads, so Nagelmackers was forced to continue using the much longer (18 hours) route through the Brenner Pass. As losses mounted, the service was reduced. With only ten cars as assets and debts far exceeding his capital, Nagelmackers disbanded the company and formed a new one under a name that was to become world famous: La Compagnie Internationale des Wagons-Lits. The company was struggling along, only slightly better off than the old one, when Nagelmackers received a letter inviting him to come to London to "discuss matters of mutual commercial interest." The invitation was from Colonel William Mann of Perkins, Ohio. Although charming and urbane, Mann was fundamentally dishonest. He once ran a tavern—profitable only because he didn't pay his bills. Later he sold stock in a non-existent oil field, collected taxes for the federal government (he turned in \$4.7 million and pocketed \$5 million), and ran several railroad-related swindles. His last misadventure in America was an attempt to compete with George Pullman using Mann-designed boudoir cars. Rather than having seats that converted into beds, each Mann car was divided into two separate sections, one for sitting and one—with real beds—for sleeping. The cars accommodated only 16 passengers, half the number who could share a Pullman, and cost more to build; thus they were quite unprofitable. Mann soon admitted defeat and took two of his cars to England where he held lavish receptions to drum up interest in them. The European railroad executives were unenthusiastic, except for young Nagelmackers who swallowed hook, line, and sinker everything Mann told him about his grandiose plans. Moreover, he even agreed to make his company a subsidiary of the Mann Boudoir Sleeping Car Company.

Nagelmackers, with boundless energy, traveled around Europe trying to sell cars to kings, nobles, and anyone else he thought could afford one. He quickly sold more than 50 cars and, in the process, became more convinced that there was a need for luxury train service across Europe. By now completely disenchanted with Mann, who was in London enjoying himself on the money that was rolling in, Nagelmackers and his financial backers bought him out for \$5 million and reinstituted the company under its old name in 1876.

Railroad executives were not convinced that the luxury service Nagelmackers described would be profitable, so they granted him only a few short-term contracts to attach some of his cars to various trains between Paris and Vienna, Paris and Cologne, Ostend and Berlin, and some other routes of less consequence. To the amazement of everyone but Nagelmackers, the cars were well patronized and frequently sold out. With some assistance from the influential King Leopold II, Nagelmackers started negotiating with many European countries, states, and kingdoms for a route from Paris to Constantinople. It took him almost a year to reach agreements with the eight national railway companies involved: the Eastern Railroad Company of France, the Imperial Railways of Alsace-Lorraine, the Kingdom of Württemberg State Railways, the Grand Duchy of Baden State Railways, the Royal Bavarian Lines of Communication, the Imperial and Royal Austrian State Railways, the Royal Rumanian Railways, and the Austrian Lloyd Shipping Company (for the last section by sea to Constantinople).

The route selected by Nagelmackers ran from Paris through Strasbourg, Munich, Salzburg, Vienna, Budapest, and Bucharest to Giurgiu on the Danube in Rumania. From there, the passengers were obliged to take a ferry across the river to Rustchuk, Bulgaria, where Nagelmackers had made a somewhat shaky arrangement for a special train to take them on to Varna (a port on the Black Sea coast). From there, a steamer took the passengers on an 18-hour sea voyage to Constantinople. Scheduled time was 81 hours and 40 minutes eastbound and 77 hours and 49 minutes westbound.

The service had no formal name, but newspapers quickly dubbed it the Orient Express, and Nagelmackers decided to make the inaugural run on October 4, 1883 under this name. The train was made up of two sleeping cars, a restaurant car, a luggage van, and a mail van. Each sleeping car had five compartments, each of which accommodated four people in comfortable bench seats by day and four berths at night. Washbasins and toilet compartments were at one end of each car. The large number of journalists invited on the inaugural run furnished many memorable accounts of the occasion. Nagelmackers spared no effort to ensure the success of the trip; elaborate food, excellent wine, beautiful crystal and linen, and impeccable service combined to create an ambience of unmistakable elegance and grandeur. In addition, many official receptions and welcomes were held along the way. Hungarian minstrels boarded the train at Szegedin and played vigorously for more than two hours without a stop. Passengers were even invited to the castle of the rather morose King of Bulgaria. Thus the most famous train of all time was put into service. After the euphoria of the inaugural trip wore off, the Orient Express settled into a regular routine. By 1884, it had become a daily service as far as Budapest and, a year later, to Bucharest with several alternative routings onward. By 1889, a link of track near Nish, Bulgaria, was finished, which meant that finally the same coaches could travel all the way from Paris to Constantinople. The route then became important for mail as well as passengers. Nagelmackers died of a heart attack on July 10, 1905, two weeks after his sixtieth birthday. He did not live to see the opening of the Simplon Tunnel, an incredibly important link through the Alps between Switzerland and Italy. Almost immediately, Wagons-Lits began to offer service between Paris and Venice. This became known as the Venice Simplon Orient Express (V-S-O-E) and is the only service still in operation today, thanks to its meticulous restoration by James Sherwood in 1982.

People often think of the Orient Express as one train running one route. In fact, the company operated more than 1000 coaches on routes throughout Europe and the Near East with such destinations as Lisbon, Madrid, Naples, Ankara, Beirut, Baghdad, Saint Petersburg, Berlin, Amsterdam, and London. From 1908 until the new communist

regime appropriated the company's cars and property in 1919, Wagons-Lits operated all the Trains de Luxe on the Trans-Siberian Railway.

During the Great War (now called World War I), many Wagons-Lits services were halted, and because of the German occupation of Belgium, the company was forced to move to Paris. Prior to the war, the Simplon service terminated in Trieste, then a part of the Austrian Empire, because Austrian authorities would not permit international trains to traverse the Empire without stopping in Vienna. However, the Treaty of Versailles, which ended the War, specified that new international through routes be opened in Italy, Trieste, and the newly created country of Yugoslavia, formerly the southern part of the Austrian Empire.

Thus, a new and much faster southern route via the Simplon Tunnel and across Italy and Yugoslavia to Constantinople was inaugurated on April 11, 1919. A few years later, the route was extended on the western end to London, creating probably the single most popular route of the Orient Express. This route was in service until the Germans occupied France during World War II, and then reinstated from the end of the war until the last run on May 19, 1977. The train ran from London to Paris, through the Mont d'Or tunnel to Lausanne and Brig, through the Simplon Tunnel to Milan, Venice, and Trieste, and into Yugoslavia. A second train, originating in Munich, met the first train in Ljubljana, and continuing cars were joined for the run to Belgrade. At Crveni Krst the train was divided, some cars being routed to Athens and the main train going to Constantinople via Sofia, Bulgaria. The trip from London to Constantinople took exactly 96 hours and the train was rarely late. While the train was occasionally slowed by snow, it generally made up the lost time on the straight runs in France and Italy. One exception was in February 1929, when the train was marooned for six days in huge snowdrifts near Cherkess Keui, just 70 miles from Constantinople. Indeed, in its long history, the Orient Express suffered very few mishaps. Minor accidents occurred in January 1901, when a locomotive jumped the rails and ran into the restaurant hall of the Frankfurt-am-Main Central Station, and in November 1911, when the Orient Express ran into a stationary freight train near Vitry-le-Francois because of incorrectly set signals. No one was seriously injured in either of these accidents. On the other hand, holdups in the Balkan areas near the end of the run occurred with disturbing frequency in the early days of the Express. As political stability increased in the '30s, however, the number of railway robberies declined. On any given run of the Orient Express one could be certain that deals were being made, plots were being hatched, and fortunes were changing hands. The booking lists in the archives of the train company read like pages from the *International Who's Who*. Royalty from Europe and the East were frequent passengers on the Orient Express as were military officers, entertainers, musicians, bankers, industrialists, bishops, and, of course, spies. The train was known to be a hotbed of intrigue and mystery.

Basil Zaharoff, the notorious arms dealer, always reserved all the seats in compartment seven for himself when he rode the train. The dancer Margaretha Gertrud Zelle, otherwise known as Mata Hari, was a frequent passenger, and Lord Kitchener traveled on the train as a young military intelligence officer gathering information about Austrian and Turkish fortifications. But we'll never know the identities of most of the spies, secret agents, saboteurs, and couriers who repeatedly crossed the continent on mysterious missions. All of which leads us to speculate and wonder: What was it really like to travel the Orient Express?

BIBLIOGRAPHY

Barsley, Michael. *The Orient Express*. London: Macdonald & Co., 1966.

Cookridge, E. H. *Orient Express*. New York: Random House, 1978.

Sherwood, Shirley. *Venice Simplon Orient-Express*. London: Weidenfeld & Nicolson, 1983.

THE PROGRAM

The Orient Express program consists of a main program, initialization section, end-game section, nine major subroutines, and nine short subroutines.

In the initialization section (Lines 180–260), variables are dimensioned, and text and numeric data are read into variables in four separate subroutines. Before the data are loaded, the program enters a loop which increments the value of the seed for the random-number generator, RN. This loop is terminated when any key is pressed (Line 140). If the value of RN is above the acceptable maximum (32767), 65535 is subtracted until the value of RN is within the range acceptable to the Randomize statement.

The last function performed in the initialization section is the shuffling of 24 integers. These are used later to determine the order in which the conversations with various passengers occur. Shuffling is an important function in many games, particularly those involving cards. In mathematical terms, shuffling is defined as a random

permutation, that is, generating the numbers from 1 to N in a random order without repetition. Consider this routine, commonly found in beginning programming texts, for shuffling 10 integers into M(1) to M(10):

```
10 FOR I = 1 TO 10
20   K = INT(1 + 10 * RND(1))
30   FOR J = 1 TO I
40     IF M(J) = K THEN 20
50   NEXT J
60   M(I) = K
70 NEXT I
```

A random number between 1 and 10 is generated in Line 20. This is then checked against all of the elements of M to see if it has already been used; if not, the current element of M, that is, M(I), is set equal to K. If it has been used, a new value of K is generated in Line 20. The routine is short, easy to understand, and horribly inefficient (the execution speed is proportional to the square of the numbers to be shuffled).

Consider this approach:

```
10 FOR I = 1 TO 10
20   M(I) = I
30 NEXT I
40 FOR I = 1 TO 9
50   K = I + INT((11 - I) * RND(1))
60   T = M(I)
70   M(K) = M(I)
80   M(I) = T
90 NEXT I
```

The program is two lines longer because of the necessity to load the array M(1) to M(10) with the integers 1 to 10 in Lines 10 to 30. However, the execution speed is much faster than the previous algorithm, as it is proportional to two times the numbers to be shuffled. The heart of the algorithm is found at Line 50. The first time through the loop, this function selects a random integer between 1 and 10; the second time through, between 2 and 10; the third time, between 3 and 10; and so on. The elements chosen are then exchanged with the values currently in those elements. If we think of the elements as cards, we could think of picking up one random card at a time from the deck, placing it on a pile, and then picking the next card. The final pile will thus be thoroughly shuffled. In the program, this subroutine is found at Lines 2760–2790.

The main program prints the scenario and then iterates through the 24 segments of the train journey. For each segment, the date is printed followed by the location and the arrival time. The scheduled arrival time is found in the data variable TA(n). The actual arrival time is determined by a random function at Line 310 and can be from 8 minutes early to 18 minutes late.

In some cases this function will produce a time like 271 or 1460. This problem is cured in the subroutine that prints out the time (Lines 3060–3090). The last two digits of the time are examined in Line 3070, and if they are found to exceed 59, one hour is added and 60 minutes are subtracted; this is accomplished by simply adding the integer 40 to the numeric value of the time. This subroutine then divides the time into strings representing hours and minutes and prints it in the correct format such as 3:11 or 15:00. The integer 10000 is added to the numeric value of the time so that the leading zeroes will be printed.

If an intermediate station is reached at night during sleeping hours, arrival and departure times are simply noted, and the program proceeds. If the stop is during waking hours, you are given a chance to get off the train and stretch your legs.

On the first segment of the trip out of London, you ask passengers to keep alert for information of value (Lines 520–560). Each segment of the trip has a series of data values which indicate the number of conversations with passengers that can take place (CN(n)), the meal to be served (ME(n)), and the potential hazards (HZ(n)). After the subroutines for each of these items are called in Lines 580–610, the program goes on to the next trip segment.

The breakfast and dinner subroutines (Lines 640–700 and 770–840) are similar. Each one asks if you are ready to go to the restaurant car for the meal and enters an endless loop until you press a key. The menu is then presented. Menu items are selected at random and printed on the screen. Items are centered by the LOCATE statement in Lines 690 and 820, which takes the length of the item, 80 (the width of the screen), divides by two, and moves the cursor to that location to start printing. At the conclusion of the meal, a loop, which waits until any key is pressed to clear the menu from the screen, is entered in the subroutine at Lines 2950–2970.

The subroutine to present the conversations you have with the passengers iterates through the conversations, if any, on the current trip segment. If a conversation is to be held, the number of that conversation, CS(CM), is selected from the previously shuffled list. In Line 910 the length of the string containing the conversation is checked. If it is fewer than 80 characters long, it is displayed on one screen line. If it is longer than 80 characters, the routine in Lines 920 and 930 starts iterating from the 79th character backward looking for a space. When one is found, the left side of the string up to that point (the first part of the conversation) is printed and, on the next line, the right side of the string from the space on is printed (the last part of the conversation).

Two subroutines for hazards—snowdrifts, bandits, and derailments—are nearly identical except for the dialog. There is a 65% chance of snow on three of the trip segments (Line 980). If it is snowing heavily, there is a 1% chance of getting stuck in a snowdrift.

If this happens, the trip is delayed for two days until the snow is cleared from the tracks. This routine could be made much more elaborate to simulate the time the train was stuck in a snowdrift for six days in 1929. However, that was an isolated incident, and every other time the train was stuck it resulted only in a delay and very little discomfort to the passengers.

On two trip segments in Bulgaria and Turkey, there is a 4% chance that bandits will board the train and rob the passengers. On the actual Orient Express robberies were an infrequent occurrence and, except for two instances in May and October 1891 when people on the train were kidnapped and held for ransom, bandits rarely hurt anyone, being content to steal money and jewelry.

On all trip segments there is a 2% chance of a derailment. In the event that this occurs, the locomotive, tender, and first mail coach will leave the track, and you will be stranded somewhere for a day until the track is repaired and a replacement locomotive obtained.

The probability of any of the above three hazards occurring in the program is much higher than it was in actuality. However, if the probabilities were reduced to the few thousandths of one percent that existed in reality, you might have to play the game tens of thousands of times to experience even a single misadventure.

The final major subroutine (Lines 1490–1690) checks the identities of the killer and defector that you input in the previous subroutine (Lines 1340–1470) with the actual identity. There are seven possible situations which can lead to any of five different eventual outcomes (see chart).

IF YOU DO THIS			THEN THIS HAPPENS	
Killer identified as	Defector identified as	Defector killed by bandits	You killed by killer	Defector wrongly arrested
Killer	Defector	No	No	No
Killer	Nobody	Yes	No	No
Nobody	Defector	No	Yes	No
Nobody	Nobody	Yes	Yes	No
Nobody	Killer	No	Yes	Yes
Defector	Nobody	Yes	Yes	No
Defector	Killer	No	Yes	Yes

The correct dialog for the three possible nasty events (defector killed, you killed, and defector wrongly arrested) can be selected with just three IF...THEN statements (Lines 1500, 1600, and 1630). If you get both identities correct, the indicator A5 is set equal to 1 so that you receive a congratulatory message when you arrive in Constantinople.

The short subroutines (Lines 2760–3140) are all self-explanatory; they simply produce pauses in program execution, make train noises, check for yes/no answers, and the like. The end-game segment (Lines 3189–3330) presents an end-game message, displays a message of congratulations if you got both identities correct, and asks if you want to ride again.

PROGRAM VARIABLES

- A Answer of user (0 = yes, 1 = no)
- A\$ Answer to string-input query, user input
- A1, A2 Answer to identity of defector and killer, user input
- A3, A4 Actual identity of defector and killer

A5 Indicates if user identifications were correct
 C\$(n) Conversations of passengers, n = 1 - 24
 CM Conversation number, index of CS
 CN(n) Number of conversations per trip segment, n = 1 - 24
 CP(n) Conversation indicator (0 = passenger, 1 = waiter, 2 = cook), n = 1 - 24
 CS(n) Conversation number, n = 1 - 24
 DA(n) Day of trip by trip segment, n = 1 - 24
 HW Hazard, derailment indicator
 HX Hazard, bandit-attack indicator
 HY Hazard delay in days
 HZ(n) Hazards on each trip segment, n = 1 - 24
 I Index indicator
 J Trip-segment indicator
 K, KA Index indicators
 LA\$(n) City, n = 1 - 24
 LB\$(n) Country, n = 1 - 24
 MB\$(n) Meal, breakfast, name of menu item, n = 1 - 13
 MD\$(n) Meal, dinner, name of menu item, n = 1 - 25
 ME(n) Meal indicator by trip segment, n = 1 - 24
 N\$(n) Names of passengers, n = 1 - 25
 RN Random seed for random-number generator
 T Time, temporary for printing
 T\$ Time, string variable, temporary for printing
 TA(n) Time of arrival, scheduled, by trip segment, n = 1 - 24
 TB Time of arrival, actual
 TD(n) Time of departure by trip segment, n = 1 - 24
 TN Time, minutes early or late
 X Temporary variable
 X\$ Temporary string variable

PROGRAM LISTINGS

[Download ORIENT.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF : ON ERROR GOTO 3160
110 LOCATE 10, 27 : PRINT "The Orient Express, 1923"
120 LOCATE 13, 28 : PRINT "(c) David H. Ahl, 1986" : LOCATE 23, 27
130 PRINT "Press any key to continue." : RN = -32768!
140 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND
150 WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN : CLS
160 PRINT TAB(22) "The Mysterious Arms Deal" : PRINT : GOSUB 1710
170 '
180 'Initialization
190 DIM C$(25), CN(25), CP(25), DA(25), HZ(25), LA$(25), LB$(25), ME(25), N$(25)
200 DIM TA(25), TD(25), CS(25), MB$(15), MD$(26)
210 GOSUB 1880 : 'Read data about journey segments
220 GOSUB 2160 : 'Read statements of travelers
230 GOSUB 2430 : 'Read names of those on the train
240 GOSUB 2530 : 'Read menu selections
250 GOSUB 2760 : 'Shuffle 24 integers for later use
260 PRINT "Press any key to call a taxi..." : WHILE LEN(INKEY$) = 0 : WEND
270 '
280 'Main program
290 FOR J = 1 TO 24 : 'Iterate through locations
300 PRINT : PRINT "February" DA(J) + 13 + HY "1923" : 'Print date
310 TN = 18 - INT(27 * RND(1)) : TB = TA(J) + TN : T = TB : IF J = 1 THEN 450
320 GOSUB 3110 : PRINT "You have arrived at " LA$(J) ", " LB$(J) " at"; : GOSUB 3060
330 IF TN > 1 THEN PRINT "just" TN "minutes late." : GOTO 360
340 IF TN < -1 THEN PRINT "almost"; TN "minutes early." : GOTO 360
350 PRINT "- right on time!"
360 IF TB > TD(J) - 2 THEN T = TB + 4 ELSE T = TD(J) : 'Make sure departure is after arrival
370 IF J = 24 THEN 3190 ELSE IF ME(J) < 4 THEN 400 : 'Is it daytime?
380 PRINT "Asleep in your compartment, you barely notice that the"
390 PRINT "departure was right on time at"; : GOSUB 3060 : GOSUB 2860 : GOTO 490
400 IF J = 23 THEN GOSUB 1340 : 'Time to identify the killer and defector?
410 PRINT "Departure is at"; : GOSUB 3060 : PRINT
420 INPUT "Would you like to get off and stretch your legs"; A$ : GOSUB 2810
430 IF A = 1 THEN PRINT "Okay, you stay in your compartment." : GOTO 470
440 PRINT "Okay, but be sure not to miss the train." : GOTO 470
450 PRINT "The taxi has dropped you at Victoria Station in London."
460 PRINT "The Orient Express is standing majestically on Track 14."

```

```

470 PRINT : SOUND 500, 15 : GOSUB 2860 : SOUND 500, 30 : PRINT "All aboard.";
480 GOSUB 2860 : PRINT "train is leaving." : GOSUB 2860
490 GOSUB 2990 : 'Train noises
500 GOSUB 2860 : IF J > 1 THEN 570 : 'First leg of trip?
510 X = 3 + INT(20 * RND(1))
520 PRINT : PRINT "You speak to some of the passengers—" N$(X) ", "
530 PRINT N$(X + 1) ", " N$(X + 2) " and others—and ask them to keep"
540 PRINT "their eyes and ears open and to pass any information—no
550 PRINT "matter how trivial—to you in compartment 13. The Channel"
560 PRINT "crossing is pleasant and the first part of the trip uneventful."
570 IF J = 23 THEN GOSUB 1490 : 'Time to identify the killer and defector?
580 IF ME(J) > 0 AND ME(J) < 4 THEN ON ME(J) GOSUB 770, 720, 640 : 'Meals
590 GOSUB 860 : 'Talk to passengers
600 IF HZ(J) > 0 THEN ON HZ(J) GOSUB 970, 1110 : 'Snow or bandits on this leg?
610 GOSUB 1220 : 'Other hazards
620 NEXT J
630 '
640 'Subroutine to serve breakfast
650 PRINT : PRINT "Breakfast is now being served in the restaurant car."
660 PRINT "Press any key when you're ready to have breakfast."
670 WHILE LEN(INKEY$) = 0 : WEND : CLS : LOCATE 3, 33 : PRINT "BREAKFAST MENU"
680 FOR I = 1 TO 4 : X = 3 * (I - 1) + 1 + INT(3 * RND(1))
690 LOCATE 4 + 3 * I, (80 - LEN(MB$(X))) / 2 : PRINT MB$(X) : NEXT
700 LOCATE 19, 20 : PRINT MB$(13) : GOSUB 2950 : RETURN
710 '
720 'Subroutine to serve lunch
730 PRINT : PRINT "An enormous buffet luncheon has been laid out in the ";
740 PRINT "restaurant car." : PRINT "Press any key when you have finished.";
750 WHILE LEN(INKEY$) = 0 : WEND : PRINT " B-U-R-P !" : RETURN
760 '
770 'Subroutine to serve dinner
780 PRINT : PRINT "Dinner is now being served in the restaurant car."
790 PRINT "Press any key when you're ready to have dinner."
800 WHILE LEN(INKEY$) = 0 : WEND : CLS : LOCATE 1, 34 : PRINT "DINNER MENU"
810 FOR I = 1 TO 7 : X = 3 * (I - 1) + 1 + INT(3 * RND(1))
820 LOCATE 2 + 2 * I, (80 - LEN(MD$(X))) / 2 : PRINT MD$(X) : NEXT
830 LOCATE 18, 24 : PRINT MD$(22) : LOCATE 20, 30 : PRINT MD$(23)
840 LOCATE 22, 32 : PRINT MD$(24) : GOSUB 2950 : RETURN
850 '
860 'Subroutine to have conversations
870 FOR K = 1 TO CN(J) : 'Iterate through conversations in this trip segment
880 GOSUB 2890 : CM = CM + 1 : 'Ring compartment buzzer and open door
890 IF CP(CS(CM)) > 0 THEN X = CP(CS(CM)) ELSE X = 3 + INT(23 * RND(1))
900 PRINT "Standing there is "N$(X) ", who tells you:" : X = CS(CM)
910 IF LEN(C$(X)) < 81 THEN PRINT C$(X) : GOTO 950 : 'If short message, print it
920 CS(RN) = CS(CM) : FOR KA = 79 TO 1 STEP -1
930 IF MID$(C$(X), KA, 1) < > " THEN NEXT KA : 'Find a space near end of line
940 PRINT LEFT$(C$(X), KA) : PRINT RIGHT$(C$(X), LEN(C$(X)) - KA)
950 NEXT K : RETURN
960 '
970 'Subroutine for snow
980 X = RND(1) : IF X > .65 THEN RETURN : '65% chance of snow
990 PRINT : PRINT "It is snowing heavily ";
1000 IF X < .01 THEN 1030 : '1% chance of getting stuck in the snow
1010 PRINT "but the tracks have been cleared and the train"
1020 PRINT "will not be delayed." : RETURN
1030 PRINT "and the train is forced to slow down." : PRINT
1040 PRINT "Oh no! The train is coming to a stop. Let's hope this is"
1050 PRINT "not a repeat of the trip of January 29, 1929 when the Orient"
1060 PRINT "Express was stuck in snowdrifts for five days." : PRINT : GOSUB 2860
1070 PRINT "But it looks like it is!" : GOSUB 2860
1080 PRINT "You are stranded for two days until a snowplow clears the track."
1090 PRINT "The train is now exactly two days behind schedule." : HY = HY + 2 : RETURN
1100 '
1110 'Subroutine for bandits
1120 IF RND(1) > .04 THEN RETURN : '4% chance of bandits
1130 IF HX = 1 THEN RETURN ELSE HX = 1 : 'Only one bandit attack
1140 PRINT : PRINT "You are rudely awakened from a deep sleep by a loud noise"
1150 PRINT "as the train jerks to a halt." : GOSUB 2890
1160 PRINT "You are shocked to see a bandit waving a gun in your face."
1170 PRINT "He demands that you give him your wallet, jewelry, and watch." : PRINT

```

```

1180 GOSUB 2860 : PRINT "The bandits are off the train in a few moments with"
1190 PRINT "their loot. They disappear into the forest. No one"
1200 PRINT "was injured, and the train resumes its journey." : RETURN
1210 '
1220 'Subroutine to deal with miscellaneous hazards
1230 IF RND(1) > .02 THEN RETURN : '2% chance of derailment
1240 IF HW = 1 THEN RETURN ELSE HW = 1 : 'Only one derailment
1250 PRINT : PRINT "You hear a loud screeching noise as the train comes to a"
1260 PRINT "crashing stop. The engine, tender, and first coach are"
1270 PRINT "leaning at a crazy angle. People are screaming." GOSUB 2860
1280 PRINT : PRINT "While not as bad as the derailment at Vitry-le-Francois in"
1290 PRINT "November 1911, there is no question that the front of the"
1300 PRINT "train has left the track." : GOSUB 2860 : PRINT
1310 PRINT "You are stranded for exactly one day while the track is"
1320 PRINT "repaired and a new locomotive obtained." : HY = HY + 1 : RETURN
1330 '
1340 'Subroutine to identify defector and killer
1350 PRINT : PRINT "The Turkish police have boarded the train. They have been"
1360 PRINT "asked to assist you, but for them to do so you will have to"
1370 PRINT "identify the killer (the dealer in machine guns) and the defector"
1380 PRINT "(the Scotch drinker) to them. The arms dealers are lined"
1390 PRINT "up as follows:" : PRINT : PRINT " (1) Austrian, (2) Turk, ";
1400 PRINT "(3) Pole, (4) Greek, (5) Rumanian." : PRINT
1410 INPUT "Who is the defector (a number please);A1
1420 INPUT "and who is the killer";A2 : GOSUB 2860
1430 PRINT : PRINT "The police take into custody the man you identified as the"
1440 PRINT "killer and provide a guard to ride on the train with the"
1450 PRINT "defector. You return to your compartment, praying that"
1460 PRINT "you made the correct deductions and identified the right men."
1470 PRINT : GOSUB 2860 : RETURN
1480 '
1490 'Subroutine to check the identities
1500 IF A1 = A3 OR A1 = A4 THEN 1600 : 'Defector saved?
1510 PRINT : PRINT "You are suddenly awakened by what sounded like a gunshot."
1520 PRINT "You rush to the defector's compartment, but he is okay."
1530 PRINT "However, one of the other arms dealers has been shot." : GOSUB 2860
1540 PRINT : PRINT "You review the details of the case in your mind and realize"
1550 PRINT "that you came to the wrong conclusion and due to your mistake"
1560 PRINT "a man lies dead at the hands of bandits. You return to your"
1570 PRINT "compartment and are consoled by the thought that you correctly"
1580 PRINT "identified the killer and that he will hang for his crimes."
1590 '
1600 IF A2 = A4 THEN A5 = 1 : RETURN : 'Killer is still on the train
1610 GOSUB 2890 : PRINT "A man is standing outside. He says, 'You made a"
1620 PRINT "mistake. A bad one. You see, I am the machine-gun dealer."
1630 IF A1 < > A4 THEN 1660 : 'Wrongly identified defector as killer?
1640 PRINT "Moreover, you incorrectly identified the man who was cooperating"
1650 PRINT "with you as the killer. So the state will take care of him. Ha."
1660 PRINT : GOSUB 2860 : PRINT "He draws a gun. BANG. You are dead."
1670 PRINT : PRINT "You never know that the train arrived at 12:30, right on"
1680 PRINT "time at Constantinople, Turkey." : GOSUB 2860 : GOSUB 2860
1690 PRINT : PRINT : GOTO 3190
1700 '
1710 'Subroutine to set the scenario
1720 PRINT " It is February 1923. The following note is received at"
1730 PRINT "Whitehall: 'If you will furnish me with a new identity and a"
1740 PRINT "lifetime supply of Scotch, I will give up my life of arms dealing"
1750 PRINT "and will provide you with much valuable information. I will be"
1760 PRINT "on the Orient Express tonight. But you must contact me before"
1770 PRINT "the train reaches Uzunkopru or that swine dealer of Maxim machine"
1780 PRINT "guns will have me killed by bandits like he did to Baron Wunster"
1790 PRINT "last month.' The note is not signed."
1800 PRINT " You, a British agent, are assigned to take the train, rescue"
1810 PRINT "the defector, and arrest the killer."
1820 PRINT " You know there are five notorious arms dealers of different"
1830 PRINT "nationalities operating in Europe under an uneasy truce as each"
1840 PRINT "deals in a different kind of weapon. But it is obvious that the"
1850 PRINT "truce has ended." : A4 = A5 : PRINT : PRINT : RETURN
1860 PRINT "Press any key to call a taxi..." : WHILE LEN(INKEY$) = 0 : WEND : RETURN
1870 '
1880 'Subroutine to read meals, conversations, hazards, day, times, location

```

```

1890 FOR I = 1 TO 24 : READ X, ME(I), CN(I), DA(I), TA(I), TD(I), LA$(I), LB$(I)
1900 NEXT I : RETURN
1910 DATA 1, 0, 0, 0, 1, 0, 1430, "London", "England"
1920 DATA 2, 1, 2, 0, 1, 1855, 1919, "Calais", "France"
1930 DATA 3, 0, 1, 0, 1, 2233, 2253, "Paris (Nord)", "France"
1940 DATA 4, 4, 0, 0, 1, 2316, 2350, "Paris (Lyon)", "France"
1950 DATA 5, 4, 0, 1, 2, 600, 620, "Vallorbe", "Switzerland"
1960 DATA 6, 0, 1, 0, 2, 700, 707, "Lausanne", "Switzerland"
1970 DATA 7, 3, 1, 1, 2, 732, 734, "Montreux", "Switzerland"
1980 DATA 8, 0, 1, 1, 2, 919, 927, "Brig", "Switzerland"
1990 DATA 9, 0, 3, 0, 2, 1005, 1025, "Domodossola", "Italy"
2000 DATA 10, 2, 2, 0, 2, 1223, 1320, "Milan", "Italy"
2010 DATA 11, 1, 2, 0, 2, 1705, 1730, "Venice (S. Lucia)", "Italy"
2020 DATA 12, 0, 1, 0, 2, 1954, 2014, "Trieste", "(Free State)"
2030 DATA 13, 0, 1, 0, 2, 2044, 2110, "Opicina", "Italy"
2040 DATA 14, 0, 2, 0, 2, 2119, 2225, "Sezana", "Slovenia"
2050 DATA 15, 4, 0, 0, 3, 21, 107, "Ljubljana", "Slovenia"
2060 DATA 16, 4, 0, 0, 3, 310, 330, "Zagreb", "Croatia"
2070 DATA 17, 3, 2, 0, 3, 900, 956, "Belgrade", "Serbia"
2080 DATA 18, 2, 1, 0, 3, 1334, 1356, "Crveni Krst", "Serbia"
2090 DATA 19, 0, 2, 0, 3, 1555, 1634, "Caribrod", "Serbia"
2100 DATA 20, 1, 2, 0, 3, 1856, 1935, "Sofia", "Bulgaria"
2110 DATA 21, 4, 0, 2, 4, 45, 120, "Svilengrad", "Bulgaria"
2120 DATA 22, 4, 0, 2, 4, 406, 445, "Pithion", "Greece"
2130 DATA 23, 3, 0, 3, 4, 505, 545, "Uzunkopru", "Turkey"
2140 DATA 24, 0, 0, 0, 4, 1230, 0, "Constantinople", "Turkey"
2150 '
2160 'Subroutine to read statements of travelers
2170 FOR I = 1 TO 24 : READ CS(I), CP(I), C$(I) : NEXT I : RETURN
2180 DATA 1, 0, "I've heard they all have different color chalets on a north-south ridge in
the Tyrol region."
2190 DATA 2, 0, "The Austrian said he likes the look of natural wood and would never paint
his chalet."
2200 DATA 3, 0, "They gave the waiter a difficult time. The Turk ordered beer and the other
four all ordered different drinks."
2210 DATA 4, 0, "The Greek told me he hunts deer, but he never hunts with any of the others
because they all hunt different animals."
2220 DATA 5, 1, "My brother delivered a case of Kirsch to the green chalet. He remembers it
being just south of the gaudy red chalet."
2230 DATA 6, 0, "The Pole asked me—can you imagine that?—if I wanted to buy any howitzers."
2240 DATA 7, 2, "One of them asked me to cook some pheasant that he shot. He said that I
should come to the yellow chalet."
2250 DATA 8, 1, "One time my brother said he delivered a case of Cognac to the middle
chalet."
2260 DATA 9, 0, "The Rumanian said he had the shortest distance to drive from his chalet to
the railroad station at Munich."
2270 DATA 10, 0, "One of them bragged that his military rifles were so accurate that he
bagged a fox with one of them."
2280 DATA 11, 0, "The man who hunts wild boar said that the pistol dealer who lives in the
chalet next to his often gives loud parties."
2290 DATA 12, 0, "The pheasant hunter complained that the arms dealer in the chalet next to
his makes far too much noise testing his mortars."
2300 DATA 13, 0, "The gin drinker bragged that he shot sixty warthogs on a single day last
August."
2310 DATA 14, 0, "The Rumanian said he looks out on a blue chalet."
2320 DATA 15, 0, "The Cognac drinker bragged that he is the best hunter and can drink more
than all of the rest of them combined."
2330 DATA 16, 0, "The one carrying the pistol said he thinks the boar's head over his
neighbor's doorway is revolting."
2340 DATA 17, 0, "One of them said that one day he'd like to lob a mortar shell at the string
of pheasants drying in his neighbor's yard."
2350 DATA 18, 0, "The Kirsch drinker said he loved the roast chicken he had to eat last
night."
2360 DATA 19, 0, "The one carrying the pistol had a second helping of pie."
2370 DATA 20, 0, "One commented that his beef dinner wasn't nearly as good as the boar that
he shot last week."
2380 DATA 21, 0, "The Pole asked for more soup."
2390 DATA 22, 0, "The one eating all the cheese mumbled that it was the same color as his
chalet."
2400 DATA 23, 0, "The Rumanian and Austrian got completely drunk last night."
2410 DATA 24, 0, "I'd like to visit the blue chalet. The owner is said to serve excellent

```



```

lobster."
2420 '
2430 'Subroutine to read the names of those on the train
2440 FOR I = 1 TO 25 : READ N$(I) : NEXT I : RETURN
2450 DATA "R. Brundt (a waiter)", "C. D'Arcy (a chef)"
2460 DATA "Herbert Hoover", "Baron Rothschild", "Guido Famadotta", "Gustav Mahler"
2470 DATA "Robert Baden-Powell", "Fritz Kreisler", "Dame Melba", "Gerald Murphy"
2480 DATA "Calouste Gulbenkian", "Captain G.T. Ward", "Sir Ernest Cassel"
2490 DATA "Major Custance", "F. Scott Fitzgerald", "Elsa Maxwell", "Mata Hari"
2500 DATA "Clayton Pasha", "Arturo Toscanini", "Maharajah Behar", "Leon Wenger"
2510 DATA "Sarah Bernhardt", "Arthur Vetter", "Isadora Duncan", "David K.E. Bruce"
2520 '
2530 'Subroutine to read menus
2540 FOR I = 1 TO 13 : READ MB$(I) : NEXT I : 'Breakfast selections
2550 FOR I = 1 TO 26 : READ MD$(I) : NEXT I : 'Dinner selections
2560 RETURN
2570 DATA "Variete Jus de Fruits", "Prunes Macerees dans le Vin"
2580 DATA "Demi Pamplermouse", "Trois Oeufs sur le Plat", "Oeufs Poches"
2590 DATA "Omelette aux Champignons", "Tranches de Pain Beurees et Confiturees"
2600 DATA "Gallettes", "Pommes-Frites", "Patisseries", "Croissants", "Yogurt"
2610 DATA "Cafe, The, Lait, Vin, Eau Minerale"
2620 DATA "Huitres de Beernham", "Cantaloup glace au Marsale"
2630 DATA "Compote des Tomates Fraiches", "Potage Reine"
2640 DATA "La Natte de Sole au Beurre", "Truite de riviere meuniere"
2650 DATA "Poulet de grain grille a Diable", "Roti de Veau a l'Oseille"
2660 DATA "Truite Saumonee a la Chambord", "Chaud-froid de Caneton"
2670 DATA "Chaudfroix des Langouste a la Parisienne"
2680 DATA "Les Noisettes de Chevreuil Renaissance", "Becasses a la Monaco"
2690 DATA "Pointes d'asperge a la creme", "Parfait de foies gras"
2700 DATA "Salade Catalane", "Truffles au Champagne"
2710 DATA "Tagliatelle de carottes et courgettes", "Souffle d'Anisette"
2720 DATA "Creme de Caramel blond", "Sorbet aux Mures de Framboisier"
2730 DATA "La selection du Maitre Fromager", "Corbeille de Fruits"
2740 DATA "Les Mignardises", "Selection du vins et liquors"
2750 '
2760 'Subroutine to shuffle 24 integers
2770 FOR I = 1 TO 23
2780 K = I + INT((25 - I) * RND(1)) : X = CS(I) : CS(I) = CS(K) : CS(K) = X : NEXT I
2790 RETURN
2800 '
2810 'Subroutine to check for yes or no answer
2820 IF LEFT$(A$, 1) = "Y" OR LEFT$(A$, 1) = "y" THEN A = 0 : RETURN
2830 IF LEFT$(A$, 1) = "N" OR LEFT$(A$, 1) = "n" THEN A = 1 : RETURN
2840 INPUT "Please enter Y for 'yes' or N for 'no.' Which is it";A$ : GOTO 2810
2850 '
2860 'Subroutine creates a short pause
2870 FOR X = 1 TO 1000 : NEXT X : RETURN
2880 '
2890 'Subroutine to ring buzzer and open door
2900 PRINT : PRINT "Your compartment buzzer rings..."
2910 BEEP : FOR KA = 1 TO 300 : NEXT : BEEP : 'Ring the buzzer
2920 PRINT "Press any key to open the door."
2930 WHILE LEN(INKEY$) = 0 : WEND : RETURN
2940 '
2950 'Subroutine to finish eating
2960 LOCATE 24, 18 : PRINT "Press any key when you have finished eating";
2970 A3 = A3 + 5 * (J + 1) - POS(X) : WHILE LEN(INKEY$) = 0 : WEND : CLS : RETURN
2980 '
2990 'Subroutine to produce train noises
3000 PRINT : PRINT "Clackety clack...clackety clack...clackety clack"
3010 IF RND(1) > .5 THEN RETURN ELSE FOR KA = 6 TO 1 STEP -1
3020 FOR I = 1 TO 4 : SOUND 130, .5 : FOR K = 1 TO 180 : NEXT : NEXT
3030 IF KA = 4 THEN GOSUB 3110
3040 FOR K = 1 TO 50 + KA * 120 : NEXT : NEXT : RETURN
3050 '
3060 'Subroutine to print time
3070 T = T + 10000 : T$ = STR$(T) : IF VAL(RIGHT$(T$, 2)) > 59 THEN T = T + 40
3080 T$ = STR$(T) : PRINT " " MID$(T$, 3, 2) ":" RIGHT$(T$, 2) " ";
3090 RETURN
3100 '
3110 'Subroutine to blow train whistle

```

```
3120 IF RND(1) > .5 THEN RETURN
3130 SOUND 500, 15 : FOR K = 1 TO 1400 : NEXT : SOUND 500, 3
3140 FOR K = 1 TO 200 : NEXT : SOUND 500, 20 : RETURN
3150 '
3160 'Error-handling subroutine
3170 A5 = ERR : RESUME NEXT
3180 '
3190 End of journey
3200 PRINT : PRINT "Your journey has ended. Georges Nagelmackers and the"
3210 PRINT "management of Cie. Internationale des Wagons-Lits "
3220 PRINT "hope you enjoyed your trip on the Orient Express, the"
3230 PRINT "most famous train in the world." : PRINT : PRINT
3240 IF A5< >1 THEN 3310 ELSE BEEP : BEEP : BEEP : X = 0
3250 PRINT "Whitehall telegraphs congratulations for identifying both"
3260 PRINT "the killer and defector correctly." : GOSUB 2860 : GOSUB 2860
3270 FOR I = 1 TO 25 : FOR K = 1 TO 100 : NEXT K : LOCATE 15, 30 : PRINT X$ : BEEP
3280 IF X = 0 THEN X$ = "CONGRATULATIONS !" : X = 1 : GOTO 3300
3290 X$ = " " : X = 0 : 'X$ = 17 spaces to erase congratulations
3300 NEXT I : PRINT : PRINT : PRINT
3310 INPUT "Would you like to ride again";A$ : GOSUB 2810
3320 IF A = 0 THEN PRINT "Okay. Good journey!" : GOSUB 2860 : RUN
3330 PRINT "Okay. So long for now." : GOSUB 2860 : KEY ON : CLS : END
```

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



THE GAME

In this simulation of Amelia Earhart's around-the-world flight attempt, you take the place of the most famous aviatrix of all time. Flying in the cockpit of a twin-engine Lockheed Electra, you face the same decisions and hazards Amelia Earhart faced in 1937.

Prior to each flight, you are given information about your physical condition, the distance to the next destination, and the current weather. Once aloft, you may encounter a variety of problems: headwinds, heavy rain, engine malfunctions, excessive fuel consumption, and navigation difficulties.

Of course, your skill as an aviator is unquestioned, but to increase your chances of survival, remember the following:

- To find out the condition of the field before each landing, you attempt to contact someone on the ground—sometimes successfully, sometimes not. Should the runway be excessively muddy, you have the option to turn back.
- In case of a malfunction, you are asked if you want to make repairs, which, of course, take time. Not making repairs, however, substantially increases the probability of a major problem in the future.
- Similarly, your engines and other mechanical components will last longer if they are maintained judiciously. Flying for more than 40 hours between engine overhauls dramatically increases the probability of malfunction.
- You must balance the relationship between fuel consumption and weight. At 150 mph, the Electra gets approximately two miles per gallon. Bad weather increases fuel consumption.

As she left New Guinea for Howland Island, Amelia Earhart flew under some of the worst conditions of the trip—as you will discover—and she had to turn back for the Gilbert Islands, a decision that ultimately led to her death. Can you learn from her mistakes and make it to Howland, Hawaii, and finally back to Oakland?

THE STORY

In the summer of 1920, on vacation from her pre-med studies at Columbia University, Amelia Earhart took a trip to California. She found the West exhilarating, but what she enjoyed most of all were the air meets, carnival-like affairs with stunt flying and barnstorming. She attended every air meet she could find and was finally rewarded with a chance to ride with the not-yet-famous barnstormer, Frank Hawks. She later recalled the flight: "By the time I had got two or three hundred feet off the ground I knew I had to fly." With some financial help from her understanding mother and against the wishes of her father, she bought her first plane, a secondhand, bright yellow Kinner Canary on her 25th birthday, July 24, 1922. She recalled: "The motor was so rough that my feet went to sleep after more than a few minutes on the rudder bar."

She did what flying she could afford for the next few years. She had dropped out of the pre-med program and taken jobs teaching English to foreign students in Boston and then doing social work at Denison House, one of America's oldest social settlements. Her salary of \$60 a month didn't permit much flying, and, in fact, she was so short of cash that she arranged to lend her plane out for demonstrations so as not to be charged hangar storage, which she said, "would have annihilated my salary."

At Denison House in May 1928, she received a phone call from publishing heir and public-relations promoter George Putnam asking if she was interested in doing something dangerous in the air. She recalled, "At first I thought the conversation was a joke and said so. Several times before I had been approached by bootleggers who promised rich reward and no danger. But the frank admission of risk stirred my curiosity. References were demanded and supplied—good references." And then Putnam dropped the bombshell that would change her life forever: "Would you like to be the first woman to fly the Atlantic?"

Amelia's reply was a prompt yes—provided the equipment was adequate and the crew capable. She went to New York that same night and met Putnam. It turned out that he was looking for a female passenger—someone with

social grace, education, charm, a pleasant appearance, and not necessarily a pilot. Still, Amelia showed Putnam her pilot's license, the first granted to a woman in the U.S. However, she came away from the meeting feeling that Putnam was not impressed with her credentials.

As it turned out, her impression was dead wrong, and three days later she was formally asked to make the flight. Indeed, Putnam was far more impressed with Amelia than he initially let on, and over the next two years he repeatedly proposed marriage, which she, just as consistently, turned down. Late in the fall of 1930, at the Lockheed factory in Burbank, California, Putnam asked Amelia to marry him. Her resistance worn down, she casually accepted, and they were married on February 7, 1931. In a departure for the 1930s, Amelia continued to use her maiden name, preferring to be called "AE." Similarly, she always called Putnam "GP."

Although AE did not pilot the Atlantic flight, she was appointed honorary captain. As it turned out, this demanded far more courage than anyone anticipated. The plane—called "Friendship"—was a Fokker trimotor seaplane that had been purchased from Commander Richard E. Byrd by Mrs. Frederick (Amy) Guest. Byrd agreed to act as technical consultant for the flight, while pilot Wilmer L. "Bill" Stultz and mechanic Louis "Slim" Gordon prepared the plane itself.

All went according to plan, and on June 3, 1928, the Friendship, carrying Stultz, Gordon, and Earhart, left Boston Harbor on the initial flight leg to Halifax, Nova Scotia. On June 5, the trimotor reached Trepassey, Newfoundland, and was readied for the transoceanic flight leg.

Unfortunately, a long spell of bad weather set in, and the trio was stranded in Trepassey for 13 frustrating days. Faced with absolutely nothing to do, Stultz gave in to his one weakness, alcohol. For the next 12 days Amelia spent as much time with him as possible playing cards, talking, taking him for long walks on the beach, and otherwise trying to distract him from the bottle.

This was not the first time she had had to deal with alcoholism. Years earlier she had helped her mother hold their family together when her father became an alcoholic. When her parents were finally divorced, it was Amelia who paid all his bills and continued to think of him as an upright and virtuous man.

When, over the North Atlantic, a weather "window" finally opened on June 17, Stultz was totally inebriated and unable to rise from his bed under his own power. Putnam later recalled that "AE did what I suppose either was the bravest or silliest act of her whole career She simply got hold of her pilot and all but dragged him to the plane. It was a fine-drawn choice. He wasn't in good shape, but perhaps—once he took off—his flying instinct, which was so sure, so complete, would come uppermost."

Stultz tried to take off three times and aborted each time when the plane failed to reach the required 50 mph for liftoff. Finally, on the fourth try, Stultz managed to reach 50 mph, in spite of the two outboard engines "coughing salt water." However, from the moment of takeoff, Stultz drew from a deep reserve of skill and resolve, keeping the Friendship on course for 20 hours and 40 minutes and making an excellent landing at Burry Port, Wales, on the morning of June 18.

As she was not the pilot, Amelia expected the trip to be nothing more than an interesting adventure, after which she would slip back into a life of social work and anonymity. However, it was she who received most of the attention at receptions in Southampton and London. There was to be no return to her old way of life.

From then on, her life became a whirlwind of publicity tours, article writing, and, occasionally, flying. Putnam frequently arranged lecture tours consisting of as many as 27 or 28 engagements in a single month, with barely enough time to get from one to another.

Although she consoled herself that she was doing the causes of both aviation and women some good, she was sorely disappointed that because of her "success," she actually had less time to fly than before. Indeed, in 1932 she confided to some of her friends in the Ninety-Nines, a women's flying group, that she felt a fraud at times because of her lack of experience. As author Vincent Loomis observed in his book, *Amelia Earhart: The Final Story*, "There was little doubt that she was sincere in wanting to promote the cause of women in aviation, but there was not much regard for her ability as a pilot and she knew it. It was time to make a true record flight."

And make it, she did. Not just one, but many. On May 20, 1932, Earhart piloted a Lockheed Vega across the Atlantic, becoming the first woman to do so. Although her goal had been to land in Paris, she flew through five wicked hours of stormy North Atlantic weather and was forced to land in a meadow near Londonderry, Ireland. Nevertheless, five years to the day after Lindberg made the first solo flight across the Atlantic, she duplicated his feat. A few months later, on August 24, 1932, she set the women's transcontinental speed record, flying nonstop from Los Angeles to Newark in 19 hours and 5 minutes.

A year later, in July 1933, she entered the Bendix east-to-west transcontinental race. She was the third competitor to finish and the first woman. Six days later, flying back to Newark, she broke her own record, making the cross-country flight in 17 hours and 7 minutes.

In January 1934, six Navy aircraft made the first Pacific crossing from the mainland to Hawaii. Amelia resolved to do the same flight—solo—as soon as possible, but a heavy lecture schedule prevented her from attempting it until a year later. On Christmas Day 1934, Amelia's faithful Vega was lashed to the aft tennis deck of a Matson cruise ship bound for Hawaii. Shortly after the first of the year, she and the plane were ready for the flight to the mainland. Unfortunately, the weather was not ready; torrential rains doused everyone's spirits for nearly a week. Finally, on the afternoon of January 11, 1935, the weather cleared enough for Amelia to slip out on what she announced was a test flight. Of course, it was no test flight and, after flying through the night and landing at Oakland, California, 2400 miles later, Amelia Earhart became the first person, male or female, to fly solo across any part of the Pacific Ocean. Now she was a legend in truth as well as in Putnam's public-relations campaigns.

Earhart made one more record flight, a 2185-mile trip from Mexico City to Newark nonstop in 14 hours and 19 minutes on May 8, 1935, before turning her attention to preparing for and financing a round-the-world flight attempt.

Amelia was increasingly uncomfortable flying single-engine planes over large bodies of water, so she decided she must have a twin-engine plane for the round-the-world attempt. Moreover, special fuel tanks and a fuel-management system would have to be fitted for the long Pacific Ocean flight legs. Also, a Sperry autopilot was to be installed to give Amelia some relief on the longer flight legs.

While Putnam was busy trying to raise money and obtain political support, Paul Mantz, AE's technical advisor, began to prepare the plane, and Amelia began as tight a flight-training regimen as she could squeeze between her lecture tours. (She gave 150 lectures in 1936.) She also had to learn to fly the Lockheed Electra, a relatively large twin-engine, 10-passenger transport plane.

In August and September 1936, Amelia, along with Mantz and mechanic Bo McKneely, made several long-distance shakedown flights. It was a good thing that they did. On one flight the fuel system didn't work properly, and the recently-installed navigation hatch blew open.

As the planning progressed, it became quite apparent that Amelia would not be able to fly the Electra and navigate at the same time. Thus it was decided that she would take along a navigator, at least for the Pacific Ocean flight legs. Harry Manning, a ship's captain who had explained the rudiments of celestial navigation to AE on the way back from England in 1928, was selected as the navigator.

Amelia decided to give Manning a small practical test of his abilities, and in early January 1937 she took him far out over the Pacific and asked him to plot a course back to Los Angeles. On the return, they hit the California coast about 200 miles north of Los Angeles. AE claimed the navigation was in error, while Manning said Amelia flew off course and tended to drift consistently to the left when trying to follow a heading. In any event, both of them agreed that an assistant navigator would be a wise precaution on the Pacific flights.

The best choice for an assistant navigator seemed to be Fred Noonan, one of Pan Am's finest navigators—at least until just two months earlier when the airline had fired him for drinking on the job. Pan Am manager Harry Drake

recalled of Noonan, "Many were the nights I carried him home and rolled him into bed dead drunk." When Noonan was sober he was one of the ablest navigators in the world, and he promised Amelia that he would stay sober for the trip.

Originally, the round-the-world flight was intended to proceed from Oakland, California in a westward direction, beginning with the three long Pacific flight legs. Preparation was finally completed in early 1937, and the takeoff planned for March 15. The crew was assembled in Oakland in early March, but as it had been so many times in the past, the weather was uncooperative.

Finally, at 4:00 P.M. on March 17, the Electra was pulled from the hangar, and at 4:37 it lifted off for Hawaii. Poor weather returned as night fell, but Amelia handled the plane well while Manning and Noonan took star sightings, manned the radio, and continually plotted and replotted the course.

As the Electra neared Hawaii, the radio operators at Makapuu asked for a radio transmission one minute long from the aircraft to provide a fix on its position. Noonan held down the telegraph key, but the generator could not deliver the power required for such a long transmission and burned out. Fortunately, a second generator powered most of the other electrical gear on the aircraft, so the loss was not too serious.

The Electra touched down at Wheeler Field 15 hours and 52 minutes and 2410 miles after leaving Oakland. Early the next morning, the Electra was flown to Luke Field, which had a longer runway. There 590 gallons of high-octane military fuel were added to her tanks, bringing to 900 gallons the total fuel on board for the 1800-mile flight to Howland Island.

At 7:35 A.M. on March 20, with Manning and Noonan aboard, AE started the long taxi down the runway. As the plane gained speed, suddenly it pulled to the right, and ten seconds later it lay in a crumpled heap on the side of the runway. Witnesses differ in their accounts of the accident. Some claim a tire blew out. Amelia believed the right shock absorber gave way. But Paul Mantz thought that AE was jockeying the throttles—something he had warned her not to do many times in practice flights.

The plane was taken by ship back to California where—\$25,000 and five weeks later—it was repaired and readied for another flight attempt. Manning decided to quit the adventure, giving as his reason that his leave time from his company was up. Much later he admitted that he had felt that "Amelia was responsible for the crash in Hawaii. She overcorrected to the left, then to the right."

The repairs delayed departure until May. That meant that AE would be making the Atlantic crossing in late June and the Caribbean flights in early July. Normal weather conditions for that period were considered unfavorable, so the direction of the flight was reversed. It would be made from west to east—from Oakland across the U.S., down to South America, across the Atlantic, across Africa and the Arabian Gulf to India, across Southeast Asia down to Australia, then to New Guinea, Howland Island, Hawaii, and back to the U.S.

Early on May 21, Amelia, Putnam, Noonan, and McKneely climbed into the Electra for another shakedown flight. Without a word to the press or anyone else, the round-the-world flight attempt was underway. The flights from Oakland to Burbank to Tucson went off without a hitch. An engine fire on the ground at Tucson caused some minor damage to the rubber fittings but was cleaned up in a few hours.

The next morning a ferocious sandstorm temporarily blocked the way out of Tucson, but the Electra finally reached New Orleans on the night of May 22. On Sunday morning, May 23, AE took off for the 688-mile flight to Miami, where she settled in for a final week of preparation. The plane was fully serviced, checked, and rechecked; long-range weather forecasts were collected; and thousands of details were attended to.

On June 1, 1937, at 5:56 A.M., Earhart and Noonan lifted off from Miami bound for San Juan, Puerto Rico. Gorgeous weather was their welcome companion, and they set down in San Juan at 1:10 p.m., right on schedule. Getting up at 3:45 A.M. on June 2, Amelia hoped for a dawn departure, but she was not able to take off until nearly 7:00 A.M. on the flight to Caripito, Venezuela. The flight was short, 624 miles, but AE had to buck 30 mph headwinds the entire way.

Heavy black rain clouds hung thick about Caripito as the Electra lifted off early on the morning of June 3. Again, strong headwinds cut the average speed, and it took nearly 4 1/2 hours to cover the 610 miles to Paramaribo, Dutch Guiana.

A very early departure from Paramaribo the next day left AE and Noonan without a current weather report; nevertheless, after some ten hours of flying, the town of Fortaleza, Brazil, 1332 miles away, came into view. A fuel-

gauge leak had to be fixed, and because Pan Am had excellent facilities there, AE decided to have an engine overhaul in Fortaleza in preparation for the Atlantic crossing, rather than at the jumping-off airport in Natal, Brazil.

The next day they took to the air at 4:50 A.M. and arrived, 270 miles later, in Natal at 6:55 A.M. The weather was unsettled all the way, and a tropical deluge caught the plane just as it landed. Amelia had hoped to leave that evening for Africa, but the rain squalls and muddy field prevented their departure. She finally got off at 3:15 A.M., using a secondary grass runway because a perverse wind was blowing directly across the longer, lighted runway. Noonan had been drinking heavily with his old Pan Am buddies, and observers at Natal sensed a growing tension between pilot and navigator that belied their outward cordiality.

Headwinds prevailed for most of the way. Then came a stretch of doldrums, some clear skies, and finally, in the words of AE, "the heaviest rain I ever saw. Tons of water descended, a buffeting weight bearing so heavily on the ship I could almost feel it." Although Dakar, French West Africa, was their objective, when they reached the coast a thick haze blanketed the landscape and there was no sign of civilization. Noonan thought that they should turn south, a correct judgment, because, as they later learned, they were 80 miles north of Dakar. However, AE decided to turn north, and half an hour later they found themselves at St. Louis, Senegal.

The following day, June 8, they flew the 163 miles to Dakar, where they were forced to lay over to repair a broken fuel gauge. AE also decided to have an overall engine check there. From June 9 to 14, they hopped across the African continent in six flights, varying in length from 340 to 1150 miles. A variety of problems were faced and overcome: weather, navigation, language, minor malfunctions, and fuel.

On June 15, the pair flew from Assab, Eritrea, to Karachi, India. A few hours out, the mixture control lever jammed, preventing AE from regulating the quantity of fuel consumed by the right engine. To economize, she reduced her speed dramatically. Nevertheless, they covered the 1920 miles in 13 hours and 10 minutes and became the first flyers ever to make a non-stop flight from the Red Sea to India.

In 1937, Karachi aerodrome, the main intermediate point for all air traffic from Europe to India, Australia, and the Far East, was one of the biggest in the world. AE and Noonan spent two days in Karachi having a major engine overhaul and replacing many small but important items for the first time on the trip.

On June 17, AE piloted the Electra 1390 miles to Calcutta, India. Despite a series of severe rain squalls, the Electra averaged 163 mph, making the trip to Calcutta in 8 1/2 hours.

Now, in the middle of the monsoon season, AE faced a number of risky situations. On June 18, the field at Calcutta was thoroughly soaked, making a takeoff very dangerous. However, there was a momentary break in the weather, and she knew that she might not get another chance to get out for several days or even weeks. She described the takeoff: "The plane clung for what seemed like ages to the heavy sticky soil before the wheels finally lifted, and we cleared with nothing at all to spare the fringe of trees at the airdrome's edge." A bit over two hours later she put down at Akyab, Burma, refueled, and took off for Rangoon. However, the weather grew increasingly hostile, until the pair found themselves in monsoon rains so savage that they beat patches of paint off the wings. After trying to get through for two hours, AE gave up and retreated to Akyab.

On June 19, the pair set out from Akyab bound for Bangkok, Siam, but again monsoon rains forced a landing at Rangoon. Horrible weather continued to plague them as they barely managed to get through the following day to Bangkok for refueling, and then on to Singapore.

Early on June 21, they flew to Bandoeng, Java, where AE decided to lay over two days to let the local KLM mechanics give the Electra a good going over. At 3:45 A.M. on June 24, as AE was warming up the plane, she found that an instrument refused to function. Repairs took a good part of the day, and they did not get off until 2:00 P.M. AE reached Saurabaya, Java, late in the day but, because of continued problems with the instruments, she was forced to return to the much better facilities at Bandoeng for more repairs the next day.

The instrument problems seemed finally cured, and on June 27, Earhart and Noonan left Bandoeng for Australia. Bucking strong headwinds most of the way, AE was forced to put down at a tiny airstrip at Koepang on the island of Timor. Early the next morning they set out across the Timor Sea, again bucking strong headwinds, and landed at Port Darwin, Australia, four hours later. There they were pounced upon by a medical inspector and quarantined on the plane for ten hours.

At 6:29 A.M. on June 29, the pair took off for Lae, New Guinea. They covered the 1200 miles over a portion of the Indian ocean dotted with small islands in 7 hours and 43 minutes. Adverse wind conditions and threatening clouds held the flyers at Lae for two days. In addition, Noonan was unable because of radio difficulties to set the chronometers, which were vital to accurate navigation.

Discouraged by these problems and steadily losing faith in Noonan because of his drinking, AE worked out a revised flight plan with the assistance of Harry Balfour, the Guinea Airways radio operator at Lae. Amelia tried unsuccessfully to persuade Balfour to go with her in addition to or instead of Noonan. The new routing was slightly north of the original course but passed over Nauru Island, which, because of its giant phosphate mining lights, was one of the few islands visible at night.

AE took off from Lae at 10:00 A.M. on July 2 and reached Nauru 11 hours later, right on schedule. From there she turned slightly south to Howland. However, her old bugaboo, consistently drifting left when following a bearing, raised its ugly head again. Thus, eight hours later when she thought she was about 100 miles out of Howland, she was indeed 100 miles short but also 170 miles north. After briefly searching and finding no sign of Howland, she made a desperate about-face in an attempt to reach the Gilbert Islands four hours to the west.

However, she was again north of her intended course, so instead of hitting the Gilberts she reached Mili Atoll, one of the southernmost atolls of the Marshall Islands chain. As she tried to put down on a long stretch of coral at Barre Island, the landing gear caught on the coral, the plane was wrenched to a stop, a wing was torn off, and Noonan was thrown forward, injuring his forehead and knee.

Earhart and Noonan were aided by the Marshallese, but word of their landing spread, and several days later they were picked up by the Japanese military, who occupied the Marshalls. On July 14, the flyers and the wrecked Electra were put aboard the Koshu, a small Japanese survey ship. On July 19 the Koshu reached Truk Island, where Earhart and Noonan were transferred to a Japanese Navy seaplane and flown to Saipan, Japanese headquarters in the Pacific.

There they were accused by the Japanese of spying, and were mercilessly questioned. Between interrogations, they were held in small damp cells in Garapan prison where, on a diet of weak soup, both became ill with dysentery. Resenting the treatment, Noonan eventually lost his temper and threw his bowl of soup at a guard. He was immediately taken out and beheaded. Amelia's strong willpower kept her going for 14 months until finally, in August 1938, she died of dysentery.

In 1935, Charles Lindberg had given up flying and moved to England after the kidnapping and death of his son. That same year Wiley Post was killed in a crash on a flight with Will Rogers in Alaska. A few years earlier Eddie Rickenbacker had given up flying to become an airline executive, and Blanche Stuart Scott had given up her role as "Tomboy of the Air" for a career in radio and the movies. Thus, with the death of Amelia Earhart in 1938, the golden age of aviation came to a close.

BIBLIOGRAPHY

Earhart, Amelia. *Last Flight*. New York: Harcourt, Brace and Company, 1937.

Loomis, Vincent V. *Amelia Earhart: The Final Story*. New York: Random House, 1985.

THE PROGRAM

The Around-the-World Flight main program is only 29 lines long (Lines 230–510); most of the calculations and dialogue are contained in eight major subroutines, 19 sub-subroutines, an initialization section, and an end-of-game summary section.

In the initialization section (Lines 160–200), variables are dimensioned and initial values are entered by means of two subroutines. I have, of course, used city and country names as they were in 1937; most of the African names are different today.

Three lines (130, 140, 200) are used to get a seed for the Randomize statement without the user having to enter a number between -32768 and 32767. The variable RN is set to -32500 and then, in Line 140 and 200, RN is incremented by 1 until any key is pressed. The length of the input buffer, LEN(INKEY\$) will be zero until a key is pressed. After a key is pressed, LEN(INKEY\$) will be greater than zero, and RN will have some value greater than -32500. Line 200 checks the value of RN to see that it is not out of range (above 32767) and, if it is, subtracts 65535 as often as necessary to bring RN into the range -32768 to 32767. It is then used as the seed for RANDOMIZE.

For the most part, the main program simply calls one subroutine after another. Very short operations are done in the main program. These include printing the current location, airplane condition, and weather; determining if there is a delay in takeoff; resetting the day and date; and fueling the plane.

The date subroutine (Lines 530–620) determines and prints the month and date based on the number of days into the flight (DY) from the starting date of May 20, 1937.

The aircraft-repairs subroutine (Lines 640–710) looks to see if there has been a malfunction in a previous flight leg ($M > 0$). If so, you are asked if you want to make repairs. (Hint: You are well-advised to do so. Amelia Earhart always made necessary repairs, even to the smallest items, as soon as possible. Not making repairs substantially increases the probability of major problems on future flight legs.) If a repair is to be made, the time (in hours) to fix it is a direct function of the facilities of the airport; a minor repair will take three hours at a well-equipped airport and nine hours at a poorly-equipped one (Line 690). If the repair time is five hours or more and the next destination is more than 600 miles away (more than four flying hours) or if the repair time is nine hours or more, you will not be able to take off the same day (Line 700).

The major-overhaul subroutine (Lines 730–830) is similar to the repairs subroutine. The program advises you to have a major overhaul "sometime soon" if flying time is between 39 and 60 hours and "as soon as possible" if flying time is more than 60 hours. You are also permitted to have a major overhaul at any of four very well-equipped airports: Miami, Fortaleza, Karachi, and Bandoeng. A major overhaul is not possible at airports with fair or poor facilities.

What is the probability of mechanical trouble? This is a function that took some thought and finally turned out to be quite simple. The mechanical condition of the plane is determined by the total number of flying hours and the speed flown. Although it is possible to push the plane to its limit, the engines and other mechanical components will last longer if they are used with, as Amelia put it, TLC. On the other hand, you can't fly too slowly, or fatigue becomes an overwhelming factor. Hence, the mechanical condition can be represented by cumulative distance times average speed. This is a better measure of wear and tear on the plane than just hours flown. Condition (PC) is normalized to 1.0 for 6000 miles at 150 mph (Line 1500), or 40 flying hours. On her round-the-world flight attempt, Amela tried to have a preventative overhaul approximately every 40 flight hours.

However, you may be tempted to fly longer intervals between overhauls or may find yourself at an airport that does not have the facilities for a major overhaul. What then? Until the recommended overhaul interval is reached, the plane can be considered very reliable. (See major engine-problem routine at Lines 1490 to 1610). At one-half the overhaul interval there is a slim 3% chance of malfunction, and even at 80% of the interval, the chance of a problem has risen to only a little over 5%. However, beyond the overhaul interval, the probability of a malfunction rises dramatically; fly 1.3 times the recommended interval and the chance of a problem is nearly 78%, while flying 1.5 times the interval increases the chance of failure to 92%. Obviously, the probability can never exceed 100%.

If we plot these probabilities (see graph on next page), we see a very familiar curve. Although it can be expressed several ways, the tangent function is one of the simplest. The upper and lower limits of the tangent function are $\pi/2$ and $-\pi/2$, but we want the probability to range between 0 and 1. Okay, that's easy to fix by dividing the answer by π and adding 0.5. However, we want the input variable to range between 0 (no flight hours) and 2 (a risk-taker flying twice the recommended interval before servicing). There are several pairs of constants that make the probability of a problem equal to 10% at the scheduled maintenance point (my estimated figure for Amelia's Lockheed Electra). I finally settled on the following equation (Line 1510) in which PC represents the plane condition (ranges between 0 to 2 or more, depending upon speed and hours flown) and MP represents the probability of a mechanical problem (varies between 0 and 1):

$$MP = \text{ATN}(14 * PC - 17) / 3.14159 + .5$$

This probability is compared to a random number between 0 and 1 (Line 1520) to determine if there is an actual failure. If there is a failure in one engine, you have a 33.3% probability of being able to nurse the plane along to the next airport on one engine, a 33.3% chance of having to go back to the airport from which you just took off, and a 33.3% chance of a forced landing (Line 1570).

The forced landing routine (Lines 1630–1720) first checks to see if you are over water; if so, you don't have a prayer. If you are over land, you have an 80% chance of surviving a crash, although the crash ends the round-the-world flight.

Minor malfunctions, of which Amelia Earhart saw many, are annoying but seldom fatal (Lines 1740–1880). On the other hand, if a previous malfunction was never fixed, the combination of two or more minor malfunctions is equivalent to losing one engine. If you have just one minor malfunction, you are given an opportunity to push on or to return to the airport from which you took off. The longer you fly with a minor malfunction, the greater the probability that it will cause major problems—up to 5% if you cover the entire flight distance. Hence, if you are less than one-third of the way to the next location, you probably should turn back. Amelia usually lived with her minor malfunctions and turned back only once on her round-the-world flight attempt, when she had serious instrument problems between Bandoeng and Saurabaya, Java. This situation is re-created in the program (Lines 2270–2320).

Another serious problem facing the flyer is that of fuel consumption and weight (Lines 1900–2020), which, of course, are interrelated. More fuel means longer flying time, but it also means more weight and higher fuel-consumption. The Electra had dramatically different fuel consumption rates depending upon its speed; this is a key calculation (Line 1910). The Electra could cruise at between 120 and 170 mph. Amelia preferred a speed of just over 150 mph whenever possible to balance fuel consumption with flight time (and pilot fatigue). It would reduce the fun of playing the game to print the exact relationships here; but at 150 mph, the Electra gets approximately two miles per gallon. Amelia also reported several serious problems with the fuel/air mixture control system. A problem in this system requires you to throttle way back because fuel consumption increases enormously.

At Lae, New Guinea, Amelia stripped nearly everything nonessential out of the Electra in order to save precious weight and increase fuel range. This is simulated in Line 1970.

Another problem related to the fuel load is that the takeoff distance is lengthened when the plane is heavier (Lines 1080–1250). This is a major factor at airports with short runways made of grass and dirt. When these fields are wet, which is frequently the case in Brazil and Southeast Asia, the plane may not be able to get off with a heavy fuel load. Indeed, in monsoon conditions or after days of heavy rain, the plane may be unable to break out of the mud at all—regardless of fuel load.

Another problem on takeoff is that of synchronizing the two engines. Amelia had serious problems with this and smashed the Electra on the takeoff from Honolulu on her first east-to-west round-the-world attempt (Lines 1200–1240).

Weather conditions affect the round-the-world flight attempt in many ways. Bad weather may prevent you from taking off at all (Lines 410–450). In the case of bad weather, you can elect to delay your takeoff for a day or more—often a wise option if a monsoon is raging outside. The weather aloft (Lines 1320–1420) can also be a headache. In the event of a monsoon, you have a 60% chance of getting through (or around) it and a 40% chance of having to turn back. In the Caribbean you will always face strong headwinds which lengthen flying times and increase fuel consumption; across the South Atlantic you face mixed weather conditions.

Weather also affects landings (Lines 2980–3200). If it has been raining and the field is soggy, you run a risk of getting a wheel stuck in the mud on a grass and dirt field. You should try to contact the tower by radio to learn the condition of the field before you land. If it is soggy, you are given the option of going back. On short flight legs this is feasible, but on longer flights you probably won't have enough fuel.

One unusual problem that Amelia faced on her round-the-world flight attempt was a navigator with a serious alcohol problem. Although Fred Noonan had supposedly gone on the wagon before the flight, he soon reverted to his old ways, particularly as the flight went on; tensions increased, and delays grew longer (Lines 920–1010 and 2040–2110). In the program, if your navigator is drunk when you are ready to take off, you have the option of waiting a day on the chance that he will be sober tomorrow. If you take off with him not fully functional, you have to rely upon dead reckoning and landmarks—feasible over land, much more difficult over water.

Seven special situations, some of which Amelia Earhart faced, are simulated in the program. The navigation was incorrect on the Atlantic crossing (Lines 2130–2200); this you must face, even if your navigator is stone sober. But it is not a serious problem; you simply hit the coast of Africa at the wrong place 95% of the time.

Out of Akyab, Burma, you will fly through a monsoon (Lines 2220–2250). Because you are following the coast, this again is not a terribly serious problem, except that you may have to land at an intermediate destination.

Already mentioned are the instrument problems on the Bandoeng to Saurabaya flight leg. Also in this part of the world, the Australian authorities refused to recognize the signature of a doctor on Amelia's vaccination records and quarantined her, Noonan, and the Electra for ten hours at Port Darwin (Lines 3220–3240).

Your most serious problem is on the flight leg from Lae, New Guinea, to Howland Island (Lines 2340–2680). Your only visible guidance comes from the arc lights at the mines at Nauru. If your navigator is functional (Noonan wasn't), you have a 30% chance of finding Howland; if not, your chance is only 2%. If you turn back for the British Gilbert Islands, you have a 1.5% chance of finding them and a 98.5% chance of winding up—as Amelia did—at the Marshall Islands.

Two situations which, unfortunately, Amelia did not face are the flight legs of Howland Island to Honolulu (Lines 2700–2770) and Honolulu to Oakland (Lines 2790–2960).

Five frequently used subroutines are found in Lines 3820 to 4010. They check for yes/no answers to questions, provide a pause, and make beeping noises to represent a radio signal, warning indicator, and alarm clock. Incidentally, if you want to speed up play of the game, you can lower the value of the FOR loop from 1500 to 1000 or 500 or even 0.

The end-of-flight summary (Lines 4030–4130) provides you with some overall statistics of your flight and that of Amelia Earhart, and gives you a chance to play the game again.

PROGRAM VARIABLES

A Answer of user (0 = Yes, 1 = No)
A\$ Answer of user (Y or N)
AB Abort flight for one day indicator
C(n) Runway construction (n = 1 - 32, index of C\$)
C\$(n) Runway construction description (n = 1 - 3)
D Distance over a flight leg
DA Day of month
DC Distance (cumulative) total
DF Distance, maximum based on speed and fuel
DJ Distance over a flight leg, temporary
DM Distance (cumulative) since last maintenance
DX(n) Distance over a flight leg (n = 1 - 32)
DY Day into flight
F Facilities (index of F\$)
F\$(n) Facilities description (n = 1 - 4)
FU Fuel for a flight leg, gallons (user input)
FX(n) Facilities by location (n = 1 - 32)
I Index variable, temporary
J Location (current) index
JA Location (destination) index
K Index variable, temporary
LA\$(n) Location, city name (n = 1 - 32)
LB\$(n) Location, state or country name (n = 1 - 32)
M\$(n) Malfunction description (n = 1 - 11)

MØ\$ Month
 M Malfunction indicator (index for M\$)
 MD Miles flown prior to a malfunction on flight leg
 ME Miles yet to fly after a malfunction
 MJ Malfunction of instruments over Java indicator
 MP Malfunction of engine probability
 MQ Malfunction not fixed indicator
 NC Navigator condition (scale of 1 to 100)
 ND Navigator condition (0 = functional, 1 = not)
 PC Plane condition (needs servicing when PC = 1.0)
 PD Plane condition based on distance and time (DM * DM / TM)
 R(n) Runway length (n = 1 - 32, index of R\$)
 R\$(n) Runway length description (n = 1 - 3)
 RF Radio frequency change indicator
 RN Random probability of finding Howland Island also Randomize seed at start of program
 S Speed over a flight leg (user input)
 SA Speed over a flight leg, actual
 SQ Speed, reduced because of fuel (user input)
 SW Speed of wind over a flight leg
 TC Flight time (cumulative) total
 TE Flight time over a flight leg, expected actual
 TF Flight time, maximum based on fuel and speed
 TG Ground time, total at one location
 TM Flight time (cumulative) since last maintenance also time in minutes at end of program
 TP Ground time for overhaul
 TQ Ground time for repairs
 TR Flight time, temporary
 W Weather, probable, at airport
 WA Weather, actual, at airport (index of W\$)
 W\$(n) Weather conditions description (n = 1 - 6)
 WX(n) Weather, probable, at each airport (n = 1 - 32)
 X\$ Temporary string variable
 X Temporary variable for random number
 Y Temporary variable
 Z\$(n) Type of crowd (n = 1 - 6)
 Note: all variables use the following measurement units:

D	Distance	Miles
S	Speed	Miles/hour (mph)
T	Time (air)	Hours
T	Time (ground)	Days

PROGRAM LISTINGS

[Download EARTHART.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF
110 LOCATE 10, 17 : PRINT "Around the World Flight of Amelia Earhart, 1937"
120 PRINT : PRINT : PRINT TAB(29) "(c) David H. Ahl, 1986" : LOCATE 23, 27
130 PRINT "Press any key to continue." : RN = -32500
140 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND : GOSUB 4150 : 'Print scenario
150 '
160 'Initialization
170 DIM LA$(35), LB$(34), FX(35), C(35), R(35), WX(35), DX(35), M$(12)
180 GOSUB 3260 : GOSUB 3450 : 'Initialize text variables and airport data
190 PRINT TAB(16) "Press any key when you're ready to go"; : WHILE LEN(INKEY$) = 0
200 RN = RN + 1 : WEND : WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN : CLS
210 '
220 'Main program
230 J = J + 1 : DY = DY + 1 : TG = 0 : GOSUB 3950 : GOSUB 3980 : 'New day & destination
240 F = FX(J) : W = WX(J) : D = DX(J) : 'Set variables for new location
250 GOSUB 530 : 'Date subroutine
260 PRINT "You are at "LA$(J)", "LB$(J)". Repair facilities are ";F$(F)".
270 PRINT "Runway is made of ";C$(C(J));" and is ";R$(R(J))" for your plane."
280 IF DC = 0 THEN 350 : 'Don't print mileage before you start
290 PRINT : PRINT "You have flown" DC "miles in total and you have flown"
300 PRINT INT(TM + .5) "hours since your last major overhaul."
310 GOSUB 640 : 'Aircraft-repairs subroutine
320 GOSUB 730 : 'Major-overhaul subroutine
330 IF TP + TQ = 0 THEN 350 : 'Any delay for repairs or overhaul?

```

```

340 TG = TG + TP + TQ : DY = DY + TP + TQ : TP = 0 : TQ = 0 : GOSUB 530
350 GOSUB 850 : 'Pilot-condition subroutine
360 GOSUB 920 : 'Navigator-condition subroutine
370 IF ND = 1 THEN 460 : 'Delay a day because navigator is drunk?
380 GOSUB 1030 : 'Next-destination subroutine
390 INPUT "How many gallons of fuel do you want in the plane"; FU
400 IF FU > 1150 THEN PRINT "Maximum capacity is 1150 gallons." : GOTO 390
410 'Actual weather may be slightly better or worse than expected
420 WA = INT(W + 1.6 * RND(1) - .3) : IF WA > 6 THEN WA = 6 ELSE IF WA < 1 THEN WA = 1
430 PRINT "Current weather : "; W$(WA) : IF WA < 3 THEN 480
440 INPUT "Do you want to wait a day for better weather (Y or N)"; A$
450 GOSUB 3820 : IF A = 1 THEN 480 : 'Abort flight because of weather?
460 AB = 0 : TG = TG + 1 : DY = DY + 1 : GOSUB 530 : 'Increase day counters
470 GOTO 350
480 GOSUB 1080 : 'Take-off subroutine
490 IF AB = 1 THEN 460 : 'Was flight aborted on takeoff?
500 GOSUB 1270 : 'In-flight subroutine
510 GOSUB 3950 : GOTO 230
520 '
530 'Date subroutine
540 IF DY < 12 THEN 560 ELSE IF DY > 56 THEN 590 ELSE IF DY > 41 THEN 570
550 MO$ = "June" : DA = DY - 11 : GOTO 580
560 MO$ = "May" : DA = DY + 20 : GOTO 580
570 MO$ = "July" : DA = DY - 41
580 PRINT : PRINT "Date: " MO$;DA", 1937" : RETURN
590 PRINT : PRINT "It's July 16 and your money has completely run out."
600 PRINT "Sorry, you were unsuccessful. Perhaps you and George Putnam"
610 PRINT "can raise enough money for a try again next year."
620 GOTO 4030
630 '
640 'Aircraft-repairs routine
650 TQ = 0 : IF M = 0 THEN RETURN : 'If no malfunctions, return to main program
660 PRINT "Your ";M$(M);" has been giving you problems."
670 INPUT "Do you want to have it repaired here (Y or N)"; A$ : GOSUB 3820
680 IF A = 1 THEN RETURN
690 M = 0 : PRINT "That will take" 2 * F + 1 "hours";
700 IF (DX(J + 1) > 600 AND F > 1) OR F > 3 THEN 710 ELSE PRINT "." : RETURN
710 PRINT " and will prevent you from leaving today." : TQ = 1 : RETURN
720 '
730 'Major-overhaul routine
740 TP = 0 : IF TM < 28 THEN 780 ELSE IF TM < 39 THEN 790 : 'Check flying hours
750 PRINT "You should probably have a major overhaul"; : IF TM > 60 THEN 770
760 PRINT "sometime soon." : GOTO 790
770 PRINT "as soon as possible." : GOTO 790
780 IF (J = 5 OR J = 9 OR J = 19 OR J = 25) AND TM > 12 THEN 800 ELSE RETURN
790 IF F > 2 THEN RETURN : 'Major overhaul not possible at this airport
800 INPUT "Do you want a major overhaul here (Y or N)"; A$ : GOSUB 3820
810 TP = 0 : IF A = 1 THEN RETURN
820 IF RND(1) > .7 THEN TP = F + 1 ELSE TP = F : '30% chance that overhaul takes
830 DM = 0 : TM = 0 : PRINT "That will take" TP "day(s)." : RETURN : 'extra day
840 '
850 'Pilot-condition routine
860 X = 10 * RND(1) : PRINT : PRINT "You are feeling";
870 IF X < 5 THEN 890 ELSE IF X < 8 THEN 900
880 PRINT "as if you could use some more sleep." : RETURN
890 PRINT "pretty good, all things considered." : RETURN
900 PRINT "fit as a fiddle and ready to go." : RETURN
910 '
920 'Navigator-condition routine
930 ND = 0 : NC = .002 * DC + 15 * TG : PRINT "Your navigator is ";
940 IF NC > 80 THEN 980 ELSE IF NC > 50 THEN 970 ELSE IF NC > 25 THEN 960
950 PRINT "well rested and ready to go." : RETURN
960 PRINT "a bit under the weather from drinking last night." : RETURN
970 PRINT "droopy and has a bad hangover." : GOTO 990
980 PRINT "drunk and barely able to walk."
990 PRINT "Do you want to wait until tomorrow and hope he will be in"
1000 INPUT " better shape (Y or N)"; A$ : GOSUB 3820 : IF A = 1 THEN RETURN
1010 ND = 1 : RETURN
1020 '
1030 'Next-destination routine
1040 IF J = 10 OR J = 21 THEN JA = J + 2 ELSE JA = J + 1

```

```

1050 PRINT "Your next destination is "LA$(JA)", "LB$(JA)", " DX(JA) "miles away."
1060 RETURN
1070 '
1080 'Take-off routine
1090 PRINT "Revving up engines...everything seems okay...rolling..."
1100 PRINT "...picking up speed and..."; : GOSUB 3950
1110 X = RND(1) : IF X > .99 THEN 1190 ELSE IF X > .98 THEN 1200 : 'Problem 2% of time
1120 Y = C(J) + R(J) + WA : 'Runway condition and weather
1130 IF Y > 9 AND X > .85 THEN 1160 : 'Monsoons and muddy runway?
1140 IF Y > 8 AND FU > 900 AND X > .6 THEN 1160 : 'Bad weather and big fuel load?
1150 PRINT "you're finally off!" : PRINT : RETURN
1160 PRINT "the wheels just won't lift out of the mud!"
1170 PRINT "Reluctantly you concede there is no chance of a takeoff today."
1180 AB = 1 : RETURN
1190 PRINT "the landing gear strut broke!" : GOTO 1220
1200 PRINT "engines aren't synchronized...plane is turning!"
1210 GOSUB 3950 : BEEP : BEEP : BEEP : PRINT
1220 PRINT "Disaster! The Electra is lying helpless on the runway with"
1230 PRINT "a broken wing, smashed engine, and structural damage just"
1240 PRINT "as in the ill-fated March 20 takeoff from Honolulu. So sorry."
1250 GOTO 4030 : 'Flight failed, exit program
1260 '
1270 'In-flight routines (weather, equipment, fuel consumption, navigation)
1280 INPUT "At what speed do you wish to fly";S
1290 IF S > 119 AND S < 171 THEN 1330
1300 PRINT "Minimum cruising speed is 120 mph; maximum is 170 mph." : GOTO 1280
1310 '
1320 'Weather-aloft routine
1330 WA = INT(WA + 1.6 * RND(1) - .3) : IF WA > 6 THEN WA = 6 ELSE IF WA < 1 THEN WA = 1
1340 PRINT "Current weather aloft is : "; W$(WA)
1350 IF J = 6 OR J = 7 THEN SW = 30 : PRINT "Strong 30+ mph headwind."
1360 IF J = 10 THEN SW = 15 : PRINT "Mixed weather...doldrums...headwinds."
1370 IF J < 20 OR J > 22 THEN 1440
1380 SW = 20 : PRINT "The plane is being buffeted about, ";
1390 IF RND(1) > .4 THEN 1420 : '60% chance of getting thru a monsoon
1400 PRINT "and you'll have to turn back."
1410 GOTO 1880 : 'Turning back
1420 PRINT "but you decide to push on."
1430 '
1440 'Compute flight data and update cumulative figures
1450 SA = S - SW : DJ = DX(JA) : TE = DJ / SA : 'Actual speed, expected time
1460 DC = DC + DJ : TC = TC + TE : 'Cumulative distance and time
1470 DM = DM + DJ : TM = TM + TE : 'Cumulative maintenance distance and time
1480 '
1490 'Major-engine-problem routine
1500 PC = DM * DM / (900000! * TM) : 'Plane condition; needs maint when pc = 1.0
1510 MP = ATN(14 * PC - 17) / 3.14159 + .5 : 'Probability of major engine problem
1520 IF RND(1) > MP THEN 1740 : 'Actual failure?
1530 M = 11 : GOSUB 3920
1540 PRINT "Right engine gauges are going crazy...major engine failure!"
1550 INPUT "Want to try to limp along on one engine (Y or N)"; A$ : GOSUB 3820
1560 IF A = 0 THEN 1630 : 'Trying to limp along on one engine
1570 X = RND(1) : IF X < .333 THEN 1600 ELSE IF X > .667 THEN 1650
1580 PRINT "No chance of making "LA$(JA)". You'll have to turn back."
1590 GOTO 1880 : 'Turning back
1600 GOSUB 3950 : PRINT "Whew! It looks as if you can nurse it along."
1610 GOTO 2040 : 'Skip minor problems and fuel-consumption routines
1620 '
1630 'Forced-landing routine
1640 IF J = 4 OR J = 10 OR J = 27 OR J > 28 THEN 1710 : 'Over water?
1650 PRINT "Going down...looking for a suitable place to land...nothing..."
1660 IF RND(1) > .2 THEN 1680 : '80% chance of surviving a forced landing
1670 GOSUB 3950 : PRINT " C R A S H ! No survivors." : GOTO 4030
1680 PRINT "maybe that small clearing..."; : GOSUB 3950 : PRINT "you made it!"
1690 PRINT "The plane is a wreck but at least you're alive."
1700 GOTO 4030 : 'It's all over
1710 PRINT "Going down...nothing but water..looking for a reef or anything..."
1720 GOTO 1670
1730 '
1740 'Minor malfunction
1750 IF RND(1) > .3 THEN 1900 : '30% chance of a minor malfunction

```

```

1760 IF N > 0 THEN MQ = M ELSE MQ = 0 : 'Previous malfunction not fixed?
1770 M = INT(1 + 10 * RND(1)) : GOSUB 3920 : PRINT : PRINT "Malfunction in the " M$(M)
1780 IF MQ = 0 THEN 1810
1790 PRINT "This combined with the previous malfunction of the " M$(MQ) " will"
1800 PRINT "create very serious problems for you." : GOTO 1570
1810 MD = INT(DJ * RND(1)) : 'Miles flown in this flight leg
1820 PRINT "You have flown" MD "miles of this flight leg. Do you want to"
1830 INPUT "push on (Y or N)";A$ : GOSUB 3820 : IF A = 0 THEN ME = DJ - MD ELSE ME = MD
1840 IF RND(1) < .05 * ME / DJ THEN 1850 ELSE 1870 : 'Up to 5% chance of going down
1850 GOSUB 3920 : PRINT "Uh oh. Fuel-feed system has malfunctioned also."
1860 PRINT "Things look pretty bad." : PRINT : GOSUB 3950 : GOTO 1640
1870 IF A = 0 THEN 1900 : 'Going on
1880 AB = 1 : J = J - 1 : JA = J : FU = .7 * FU : 'Turning back
1890 '
1900 'Fuel consumption
1910 TF = FU * (5.6 / S - .02) : 'Flying time for amount of fuel
1920 IF TF * .8 > TE THEN 2040 : 'Enough fuel for flight leg
1930 IF S < 121 THEN PRINT "Fuel consumption seems very high..." : GOTO 1990
1940 PRINT "You're going to be tight on fuel. Perhaps you should throttle"
1950 INPUT "back. What speed would you like";SQ
1960 IF SQ < 120 THEN INPUT "Too slow. Now then, what speed";SQ : GOTO 1960
1970 IF J > 28 AND FU > 1100 THEN 2040 : 'Longer range on stripped plane
1980 IF S - SQ < 9 THEN 2000 : 'Cut back speed enough to make a difference?
1990 IF TF * .96 > TE THEN 2040 : 'Run out of fuel?
2000 PRINT "Uh oh...the right engine is sputtering..."
2010 GOSUB 3950 : PRINT "And now the left engine too. You're out of fuel."
2020 GOTO 1630 : 'Go to forced-landing routine
2030 '
2040 'Navigation
2050 IF NC < 51 THEN 2090 : 'Is navigator functional?
2060 PRINT "Your navigator isn't going to be of much use to you today."
2070 PRINT "You'll have to rely upon dead reckoning and landmarks."
2080 IF M = 5 OR M = 7 THEN PRINT "Moreover, your " M$(M) " is on the fritz."
2090 TR = INT(TE) : IF TR < 2 THEN TR = 1.2
2100 PRINT : PRINT "You have been flying for over" TR "hours but there is"
2110 PRINT "no sign of " ; : IF AB = 0 THEN 2130 ELSE AB = 0 : GOTO 2990
2120 '
2130 'Special situations
2140 IF J < >10 THEN 2220 : 'Atlantic crossing
2150 PRINT "land. Pushing onwards. " ; : GOSUB 3950 : PRINT "Wow! Land! Look!"
2160 PRINT "Approaching coast of Africa; ahead of you is " ;
2170 IF RND(1) > .95 THEN PRINT "Dakar! Nice flying!" : J = 11 : GOTO 3010
2180 PRINT "nothing but jungle. Turning north." : GOSUB 3950
2190 PRINT "A half hour later you sight St. Louis, Senegal, and decide to land."
2200 JA = 11 : DX(12) = 163 : GOTO 3010 : 'Distance between Dakar and St. Louis
2210 '
2220 IF J < >21 THEN 2270 : 'Moonsoons out of Akyab
2230 PRINT "anything except the deluge of water. You'll have to put down at"
2240 PRINT "Rangoon...if you can find it." : GOSUB 3950 : PRINT "Look! There!"
2250 JA = 22 : GOTO 3000 : 'Reset destination to Rangoon
2260 '
2270 IF J < >25 THEN 2340 : 'Serious instrument problems in Java
2280 IF MJ = 1 THEN 2990 : 'Have we been through this already?
2290 PRINT "civilization. Moreover, several of your instruments"
2300 PRINT "are behaving quite badly. Reluctantly, you turn back to"
2310 PRINT "Bandoeng because you know that facilities at Saurabaya are minimal."
2320 DC = DC - 300 : DM = DM - 300 : TC = TC - 2 : TM = TM - 2 : J = 24 : JA = 24 : MJ = 1 :
GOTO 3040
2330 '
2340 IF J < >29 THEN 2700 : 'Lae to Howland Island
2350 PRINT "land. You spotted the arc lights at Nauru 8 hours ago."
2360 PRINT "Calling Coast Guard cutter Itasca..."; : GOSUB 3880 : GOSUB 3950
2370 IF RF = 1 THEN 2390 ELSE PRINT "Nothing."
2380 PRINT "Switch radio frequency...try again..." : RF = 1 : GOTO 2360
2390 PRINT "Still nothing." : GOSUB 3950 : PRINT "You're very low on fuel!"
2400 PRINT "You can search for Howland or turn back to the Gilbert Islands."
2410 INPUT "Want to search (Y or N)";A$ : GOSUB 3820 : IF A = 1 THEN 2470
2420 FOR K = 1 TO 4 : PRINT "Searching..." : GOSUB 3950 : NEXT K : GOSUB 3950
2430 IF NC < 30 THEN RN = .3 ELSE RN = .02 : 'If navigator okay, 30% chance of
2440 ' finding Howland Island, otherwise only 2%
2450 IF RND(1) > RN THEN 1630 : 'Go to forced-landing routine

```



```

2460 PRINT "My gosh! There it is! A tiny speck of land. WOW!" : GOTO 3010
2470 PRINT "Tuvalu, the only island in the Gilberts with a landing strip,"
2480 PRINT "is almost 4 hours distant on a course almost due west."
2490 FOR K = 1 TO 4 : PRINT "Flying..." : GOSUB 3950 : NEXT K : GOSUB 3950
2500 PRINT : PRINT "Look! Coral reefs. A small island." : GOSUB 3950
2510 PRINT "Virtually no fuel left...both engines sputtering...try to put"
2520 PRINT "it down in that flat area along the beach." : GOSUB 3950
2530 PRINT "You made it down...a wing tore off the plane...navigator injured."
2540 GOSUB 3950 : PRINT "Men in uniform are coming over the sand dunes."
2550 GOSUB 3950 : GOSUB 3950 : PRINT : IF RND(1) > .985 THEN 2650
2560 PRINT "They're Japanese. An English-speaking native tells you that"
2570 PRINT "this is Mili Atoll in the Marshall Islands. You are put on a"
2580 PRINT "warship bound for Majuro. Days later you are put on another"
2590 PRINT "Japanese warship bound for Saipan. The Japanese accuse you"
2600 PRINT "of being a spy, torture you, and put you in a tiny prison cell."
2610 GOSUB 3950 : PRINT : GOSUB 3950
2620 PRINT "After months in a tiny, damp prison cell you contract dysentery."
2630 PRINT "Your navigator is executed and in August 1938 you die of disease"
2640 PRINT "and thus become the first U.S. casualties of World War II."
2650 GOTO 4030
2660 PRINT "They're British. You're safe. In three days the USS Itasca"
2670 PRINT "picks you up and deposits you in Honolulu a week later."
2680 GOTO 4030
2690 '
2700 IF J< >30 THEN 2790 : 'Howland to Honolulu
2710 PRINT "the Hawaiian Islands. But you're buoyed by the thought"
2720 PRINT "that you found Howland Island in the middle of the Pacific."
2730 FOR K = 1 TO 4 : PRINT "Flying..." : GOSUB 3950 : NEXT K : GOSUB 3950
2740 PRINT : PRINT "Calling Honolulu...come in please" : GOSUB 3880 : GOSUB 3950
2750 PRINT "Honolulu to Electra: You're right on course. Weather is"
2760 PRINT "excellent. You should sight Diamond Head very soon." : GOSUB 3950
2770 PRINT : PRINT "Yes...there it is. What a welcome sight!" : GOTO 3040
2780 '
2790 IF J< >31 THEN 2990 : 'Honolulu to Oakland
2800 PRINT "the mainland. But you're confident you'll make it."
2810 GOSUB 3950 : PRINT : PRINT "You've been flying nearly 20 hours and you're"
2820 PRINT "bone tired. You wish your navigator could relieve you." : GOSUB 3950
2830 GOSUB 3880 : PRINT "Oakland calling Electra...Oakland calling Electra..."
2840 INPUT "Are you okay...please respond...are you okay";A$ : GOSUB 3820
2850 IF A = 0 THEN PRINT "Oakland : Glad to hear it." : GOTO 2870
2860 PRINT "Oakland: Sorry to hear that. Keep going. Just a short way now."
2870 GOSUB 3950 : PRINT " Oh yes, G.P. sends greetings." : GOSUB 3950 : PRINT
2880 PRINT "And there it is; the Pacific coast and the Golden Gate Bridge."
2890 PRINT "What a beautiful sight! Coming into Oakland...steady...steady."
2900 PRINT "Touchdown...slowing down...HUGE crowds all around...stopping."
2910 FOR K = 1 TO 4 : GOSUB 3950 : NEXT K : CLS : X = 0
2920 FOR I = 1 TO 30 : FOR K = 1 TO 100 : NEXT K : LOCATE 10, 30 : PRINT X$ : BEEP
2930 IF X = 0 THEN X$ = "CONGRATULATIONS !" : X = 1 : GOTO 2950
2940 X$ = " " : X = 0
2950 NEXT I
2960 PRINT : PRINT : PRINT : PRINT : GOTO 4050
2970 '
2980 'Landing routine
2990 PRINT LA$(JA) ". Flying on..." : GOSUB 3950 : PRINT "Look to the right!"
3000 PRINT "It looks like...yes it is...an aerodrome. What a welcome sight."
3010 PRINT "Field looks " R$(JA) " for the plane."
3020 IF M< >7 THEN 3040 ELSE PRINT "Radio broken. You'll have to try to land"
3030 PRINT "without establishing contact." : PRINT : GOTO 3090
3040 PRINT "Electra calling control tower..." : GOSUB 3880 : IF RND(1) < .1 THEN 3110
3050 PRINT "Tower to Electra...tower to Electra..." : IF WX(JA) > 3 THEN 3080
3060 INPUT "Condition of field is fine. Do you want clearance to land";A$
3070 GOSUB 3820 : IF A = 0 THEN 3130 ELSE PRINT "Repeat: "; : GOTO 3050
3080 INPUT "Field is a bit soggy. Do you want clearance to land";A$
3090 GOSUB 3820 : IF A = 0 THEN 3130 ELSE INPUT "Do you want to turn back";A$
3100 GOSUB 3820 : IF A = 0 THEN 1880 ELSE PRINT "Repeat : "; : GOTO 3050
3110 INPUT "Can't establish contact. Do you want to land"; A$ : GOSUB 3820
3120 IF A = 1 THEN PRINT "Circling...circling...trying radio again." : GOTO 3040
3130 PRINT "Coming in...steady...steady..." : GOSUB 3950 : PRINT "touchdown."
3140 IF C(JA) + R(JA) > WX(JA) > 9 AND RND(1) < .15 THEN 3150 ELSE 3170
3150 PRINT "Field is soggy...one wheel caught in mud...plane is tipping."
3160 GOTO 1220

```

```

3170 PRINT "Slowing down...turning...bring it to a stop...shut down engines."
3180 IF JA = 11 OR JA = 2 THEN K = 6 ELSE K = INT(1 + 4.9 * RND(1)) : 'Type of crowd
3190 PRINT "A " Z$(K) " crowd is waiting for you. Nice job."
3200 IF J< >28 THEN RETURN : 'Are you in Darwin?
3210 '
3220 PRINT "Australian authorities claim that your medical papers are"
3230 PRINT "not in order and hold you on the plane for 10 hours. That"
3240 PRINT "costs you an extra day." : DY = DY + 1 : TG = TG + 1 : RETURN
3250 '
3260 'Subroutine to put verbal data into constants
3270 FOR I = 1 TO 4 : READ F$(I) : NEXT I
3280 FOR I = 1 TO 3 : READ C$(I) : NEXT I
3290 FOR I = 1 TO 3 : READ R$(I) : NEXT I
3300 FOR I = 1 TO 6 : READ W$(I) : NEXT I
3310 FOR I = 1 TO 11 : READ M$(I) : NEXT I
3320 FOR I = 1 TO 6 : READ Z$(I) : NEXT I
3330 DATA "excellent", "good", "fair", "poor"
3340 DATA "concrete", "packed gravel", "grass and dirt"
3350 DATA "plenty long", "of adequate length", "barely long enough"
3360 DATA "clear", "scattered clouds", "overcast", "light rain"
3370 DATA "wind and heavy rain", "high wind and monsoon rains"
3380 DATA "thermocouple", "turn & bank indicator", "fuel gauge", "altimeter"
3390 DATA "Bendix radio direction finder", "Sperry Gyro Pilot", "radio"
3400 DATA "mixture control lever", "hydraulic system", "electrical system"
3410 DATA "engine"
3420 DATA "small", "large", "noisy", "clamorous", "restless", "tiny"
3430 RETURN
3440 '
3450 'Airport location, repair facilities, runway construction,
3460 ' runway length, likely weather, miles from last airport
3470 FOR I = 1 TO 32 : READ X, LA$(I), LB$(I), FX(I), C(I), R(I), WX(I), DX(I) : NEXT I
3480 RETURN
3490 DATA 1, "Oakland", "California", 1, 1, 1, 1, 0
3500 DATA 2, "Burbank", "California", 2, 1, 1, 1, 332
3510 DATA 3, "Tucson", "Arizona", 2, 1, 1, 1, 456
3520 DATA 4, "New Orleans", "Louisiana", 1, 1, 1, 1, 1287
3530 DATA 5, "Miami", "Florida", 1, 1, 1, 1, 688
3540 DATA 6, "San Juan", "Puerto Rico", 2, 2, 2, 1, 1053
3550 DATA 7, "Caripito", "Colombia", 1, 1, 1, 3, 624
3560 DATA 8, "Paramaribo", "Dutch Guiana", 3, 3, 1, 2, 610
3570 DATA 9, "Fortaleza", "Brazil", 1, 3, 2, 5, 1332
3580 DATA 10, "Natal", "Brazil", 3, 3, 2, 4, 270
3590 DATA 11, "St. Louis", "Senegal", 4, 3, 2, 2, 1992
3600 DATA 12, "Dakar", "French West Africa", 2, 2, 2, 1, 1974
3610 DATA 13, "Gao", "French Sudan", 3, 3, 2, 2, 1150
3620 DATA 14, "Fort Lamy", "Chad", 4, 3, 3, 1, 1027
3630 DATA 15, "El Fasher", "Fr. Equatorial Africa", 4, 3, 3, 1, 679
3640 DATA 16, "Khartoum", "Anglo Egyptian Sudan", 4, 2, 3, 1, 494
3650 DATA 17, "Massawa", "Abyssinia", 3, 3, 2, 1, 442
3660 DATA 18, "Assab", "Eritrea", 3, 2, 2, 1, 340
3670 DATA 19, "Karachi", "India", 1, 1, 1, 3, 1920
3680 DATA 20, "Calcutta", "India", 2, 2, 2, 6, 1390
3690 DATA 21, "Akyab", "Burma", 3, 2, 2, 6, 338
3700 DATA 22, "Rangoon", "Burma", 3, 2, 2, 6, 330
3710 DATA 23, "Bangkok", "Siam", 2, 2, 2, 5, 365
3720 DATA 24, "Singapore", "Asia", 2, 2, 2, 2, 895
3730 DATA 25, "Bandoeng", "Java", 1, 3, 1, 2, 635
3740 DATA 26, "Saurabaya", "Java", 4, 3, 3, 2, 365
3750 DATA 27, "Koepang", "Timor", 4, 3, 3, 2, 1148
3760 DATA 28, "Port Darwin", "Australia", 2, 1, 1, 1, 517
3770 DATA 29, "Lae", "New Guinea", 3, 3, 2, 2, 1196
3780 DATA 30, "Howland Island", "Pacific", 4, 2, 2, 2, 2556
3790 DATA 31, "Honolulu", "Hawaii", 1, 1, 1, 1, 1818
3800 DATA 32, "Oakland", "California", 1, 1, 1, 1, 2420
3810 '
3820 'Check for yes or no answer
3830 IF A$ = "Y" OR A$ = "y" THEN A = 0 : RETURN
3840 IF A$ = "N" OR A$ = "n" THEN A = 1 : RETURN
3850 PRINT "Don't understand your answer of ";A$; "."
3860 INPUT "Please enter Y for 'yes' or N for 'no.'";A$ : GOTO 3820
3870 '

```

```

3880 'Radio-signal routine
3890 FOR I = 1 TO 4 : X = 1 + 3 * RND(1) : FOR K = 1 TO X : BEEP : NEXT K
3900 FOR K = 1 TO 500 : NEXT K : NEXT I : RETURN
3910 '
3920 'Warning-beeper routine
3930 PRINT : FOR I = 1 TO 3 : BEEP : BEEP : FOR K = 1 TO 500 : NEXT K : NEXT I : RETURN
3940 '
3950 'Pause routine
3960 FOR I = 1 TO 1500 : NEXT I : RETURN
3970 '
3980 'Alarm-clock routine
3990 PRINT : FOR I = 1 TO 7 : BEEP : NEXT I
4000 PRINT "There goes the alarm. It's" INT(3 + 3.7 * RND(1)) "a.m. Y - A - W - N"
4010 FOR I = 1 TO 250 : NEXT I : FOR I = 1 TO 7 : BEEP : NEXT I : RETURN
4020 '
4030 'End-of-flight summary routine
4040 GOSUB 3950 : PRINT : PRINT : PRINT "Sorry your flight was unsuccessful."
4050 PRINT : PRINT "You flew" DC "miles and were aloft for";
4060 T = INT(TC) : TM = INT(60 * (TC - T)) : PRINT T "hours and" TM "minutes."
4070 PRINT "Your flight started on May 20 and ended on" MO$; DA " ", 1937."
4080 PRINT : PRINT "Amelia Earhart flew approximately 27,000 miles between"
4090 PRINT "May 20 and July 2, 1937 before going down at Mili Atoll"
4100 PRINT "in the Japanese-held Marshall Islands." : GOSUB 3950 : PRINT
4110 PRINT : INPUT "Would you like to try again (Y or N)";A$ : GOSUB 3820
4120 IF A = 0 THEN PRINT "Okay. Good luck!" : GOSUB 3960 : CLS : RUN
4130 PRINT "Okay. So long for now." : GOSUB 3960 : KEY ON : CLS : END
4140 '
4150 'Subroutine to print the scenario
4160 CLS : PRINT TAB(20) "Around the World Flight Attempt" : PRINT
4170 PRINT " In this simulation, you take the role of Amelia Earhart in her"
4180 PRINT "attempt to fly around the world in a twin-engine Lockheed Electra."
4190 PRINT " Prior to each flight leg, you are given information about your"
4200 PRINT " physical condition and that of your navigator, the distance to your"
4210 PRINT "next destination, and the current weather. As pilot, you must make"
4220 PRINT "many decisions before taking off, while aloft, and prior to landing."
4230 PRINT " Under ideal conditions, at 150 mph, your plane can fly 2.3"
4240 PRINT "miles on one gallon of fuel, but conditions are seldom ideal."
4250 PRINT "The Electra can hold up to 1150 gallons of fuel."
4260 PRINT " Your engine and mechanical components will last longer if they"
4270 PRINT "are maintained regularly; on the Electra, the recommended interval"
4280 PRINT "for a major overhaul is 40 hours. But remember, not all airports"
4290 PRINT "are equipped to service your aircraft."
4300 PRINT " If you have malfunctions along the way, you may want to have"
4310 PRINT "them fixed at a secondary aerodrome. Of course, this costs time."
4320 PRINT " You navigator has a serious alcohol problem. As long as your"
4330 PRINT "ground time is minimal, he won't have much chance to get lost in"
4340 PRINT "the bottle, but if you get trapped on the ground by a series of"
4350 PRINT "tropical storms and he gets drunk, you may find you have to rely"
4360 PRINT "on dead reckoning and landmarks when you get back in the air."
4370 PRINT : RETURN

```

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



THE GAME

In Tour de France, you are a bicycle racer whose objective is to win the famous three-week bicycle road race around France. The competition is fierce in this most grueling of all races: You must overcome physical pain, mechanical difficulties, rough terrain, and other problems.

Prior to the race, you are asked some questions about your physical condition and training. You are also given a chance to pedal your bicycle on a short practice run. To "pedal" your bicycle, strike two keys alternately on your keyboard. You can use the index finger of each hand or the index and middle finger of one hand. A well-practiced rider in excellent condition can pedal a bicycle at about 80 rpm.

Your bicycle is a 12-speed with ring gears (pedal) of 52 and 40 teeth, and cog gears (back wheel) of 13, 15, 17, 19, 21, 23, and 25 teeth. (See Table 1.) In your highest gear, 52/13, you will travel 8.61 meters for each pedal revolution, while in your lowest gear, 40/25, you will travel 3.45 meters. Your high gears are best for the flat, open road, while your low gears are necessary for climbing the steep mountains.

Cog Gear	13	15	17	19	21	23	25
----------	----	----	----	----	----	----	----

Ring Gear							
-----------	--	--	--	--	--	--	--

52	8.61	7.45	6.59	5.88	5.32	4.87	
----	------	------	------	------	------	------	--

40	5.75	5.06	4.55	4.09	3.75	3.45	
----	------	------	------	------	------	------	--

TABLE 1. Meters progressed for ring/cog combinations

Here are a few tips to help you place among the top finishers:

- Always use the highest gear possible for the terrain you encounter.
- You can have the computer do some of your pedaling, but it is an inconsistent performer, so you may be better off doing your own pedaling.
- There is a sprint at the end of each stage. The computer counts down the kilometers starting 10 km from the end of the stage. You must decide when to start your sprint. Start too soon and you may tire before you reach the finish line; start too late and you will not gain much time.

The Tour de France is probably the toughest athletic contest in the world. One quarter of the starters never finish the race. But the others—despite collisions, concussions, gashes, and even broken bones—continue to compete, for simply to finish the Tour is a feat any cyclist can be proud of.

THE STORY

What is the most popular sporting event in the world? The World Series? Wimbledon? The Olympic Games? Hardly. The World Soccer Cup? Getting warmer. The Indianapolis 500? Almost (it *is* the most popular one-day event). The Tour de France? Bingo! Every year, more than 20 million fans line the roads to catch just a quick glimpse of the bicycles flashing by in this most competitive race.

The Tour, which lasts three weeks, is a race of at least 4000 kilometers (2500 miles) winding through France, the Pyrenees, and the Alps, to the traditional finish line on the Champs-Élysées in Paris. The riders who compete in the Tour de France are among the toughest and most courageous athletes in the world; they have to be to survive the formidable daily pace, full-speed sprints, 60 mph spills, treacherous weather, and fickle equipment.

The Tour de France was created in 1903 by Henri Desgranges as a promotional device to sell newspapers, and today it is still owned by *l'Equipe*, a sports paper. But over the years, the Tour has become practically a sovereign state with its own motorcycle police force, its own traveling bank, and more than 2000 full-time participants (officials, mechanics, trainers, doctors, chauffeurs, technicians, salesmen, reporters, and photographers) in addition to the 170 or so riders.

Although it is theoretically possible to enter the Tour de France as an individual, it is rarely done, not only because of the high cost, but because of the support personnel and equipment required. Thus, in recent years, all of the entries have been teams—each one consisting of ten competitors and a veritable army of support troops.

Many other countries have bicycle tours, but the majority of them are relatively unimportant to the professional teams because of length (too short), time of year (before or after the main European season), or location (Africa, Asia, and East Europe pose unmanageable logistical problems). Thus, only three tours remain significant—those of Spain, Italy, and France—and, of these three, the Tour de France reigns supreme.

The route of the Tour changes each year and, for a price, a small town or village can buy a few moments of recognition on European television by becoming an evening stopover point—the end of one stage and the beginning of the next. In these small towns, shops and banks close, and all of the townspeople turn out to welcome the riders as they struggle in at the end of the daily stage. Even more exciting is the half hour in the morning before the departure on the next stage: Young and old turn out in full racing regalia to glide on their bicycles among the riders, children seek autographs, and riders chat with their fans.

The price a town must pay to start the Tour is substantial—about a half million dollars. Therefore the race usually starts in a major city, not necessarily in France. Frankfurt was the starting city in 1980, Nice in 1981, and Basel in 1982. More recently, the organizers have been promoting the idea of a region—rather than a city—hosting the race start. Thus in 1983 the race began in the southeast of Paris, Val-de-Marne Department, and in 1984 the race began north of the capital in the Seine-Saint-Denis Department.

As the route of the Tour changes from year to year, so does the length. It is generally about 4000 kilometers long, although in 1968 it was stretched to a staggering 4664 kilometers. Sometimes the route is continuous, with each new stage starting at the end of the preceding stage, although in recent years there have been some notable breaks, including a 300-kilometer train trip between Stages 22 and 23 in the 1984 race.

Generally, one stage is run each day, although this is not a hard-and-fast rule. Occasionally, a time-trial stage may be run after one of the shorter flat stages. During time trials, racers compete against the clock, starting at one-minute intervals and race completely on their own to the finish line. The time trials are regarded as particularly painful events because a rider cannot use the slipstream or draft effect of his competitors, and because tactics are secondary to pure concentration and sheer physical effort.

Sometimes there may be one or two rest days—usually just one which comes two-thirds of the way through on July 14, Bastille Day. Distances to be covered in each stage are quite different, ranging from 350 kilometers in a flat stage to as little as 20 kilometers in a time trial or one of the high mountain stages.

Although most time trials are held over fairly short distances, there are some that range longer in length. One of the most prestigious long distance events is the Hour Record time trial. In this event a racer is challenged to cover the maximum amount of distance possible in one hour. Points are awarded for winning and placing in all the time trials and race stages. Thus, for determining the overall Tour winner (calculated by a complex formula including total elapsed time and points for stage wins), time trials loom as important events.

To get a feel for an actual race, let's accompany the riders in the 1984 Tour. Seventeen teams entered the 1984 race, each with ten riders. A short prologue stage was held on Friday, June 29, two days before the official start. This was a time trial held near Paris that took the winner, Bernard Hinault, only 6 minutes, 39 seconds to complete. The tenth place rider was only 14 seconds behind the leader, and most riders agreed that this was just a small warm-up and not indicative of the race to follow. Nevertheless, Hinault got to wear the coveted yellow jersey (awarded to the winner of each stage) for the start of the actual race.

The first "real" stage of the 1984 Tour started from Seine-Saint-Denis on Sunday, July 1. The route wound 249 kilometers over secondary roads towards Louvroil in the Nord Department. The day was hot, and would have been oppressively so except for a headwind that was strong enough to cool down the passion for breakaways. Moreover, the road was narrow, rarely allowing more than three men abreast; thus, escape from the pack was effectively blocked. During this stage, the riders seemed most concerned with staying out of each other's way and avoiding the grazing contacts that can cause mass spills. Nevertheless, on a tight corner at the 78th kilometer, Jean-Rene Bernaudeau fell, scraping a knee. Before the day was over, six other riders crashed, injuring shoulders, legs, arms, and heads.

Jaime Vilamajo was fined 75 francs and penalized 10 seconds for receiving a bag of food at kilometer 158, 16 kilometers beyond the feeding zone, a designated stretch where riders can grab a light sack of food. Another rider was fined 50 francs for stopping on the left side of the road, instead of the regulation right, when he had a flat tire. In the old days, each rider carried a spare tire strapped to his back. However, as it took about three minutes for a rider to change his own tire and it takes a team mechanic only 30 seconds, today a rider with a flat simply waits at the side of the road for his team mechanic to come along and change the tire. These days, riders don't even dismount but simply stand, straddling the frame of their bicycle, while a tire is changed or a minor mechanical problem is fixed by a mechanic.

The second day of the 1984 race consisted of two stages, a 51-kilometer team time trial and an 83-kilometer stage through predominantly flat country near the Belgian border. During the afternoon stage, there was a strong sidewind, and Belgian rider Adri van den Haute, running 96th in the standings, perhaps sparked by the proximity to his native land, made a strong breakaway and built an astounding 6 minute and 23 second lead. Crowds love a breakaway, and a great cheer greeted him as he entered Douai, about midway through the stage. However, as the kilometers wore on and the sidewind turned into a rainy headwind, his lead slowly diminished, and he finally finished at Bethune with a 1 minute, 3 second lead.

The next day in Stage 5, Paulo Ferreira, an unknown Portuguese rider, unexpectedly broke away at kilometer 3, just after the start of the race. Two other riders followed him and the three of them swapped the positions of setting the pace and riding the slipstream for the entire 207 kilometers of the stage from Bethune to Cergy-Pontoise. As their lead from the pack built up through 11 minutes, 19 minutes, and ultimately 27 minutes, they experienced the exhilaration of looking back over the small hills and seeing nobody following. Until this stage, there had never been more than 13 seconds difference between the total elapsed time of the first- and fourth-place riders. Now, three completely new riders were in the top three places—Vincent Barteau, Maurice Le Guilloux, and Ferreira—and there was more than a 17-minute gap to the former first place rider who had slipped to fourth place.

Stage 6 was a 202-kilometer run over flat country to Alençon. The stage was largely uneventful until, just ten kilometers from the end, two riders in the middle of the pack locked wheels, causing a mass crash of at least 30 other riders. "What's awful about bicycling," said Greg LeMond, the great American cyclist, "is that you're supposed to get up, get back on your bike, and finish. In any other sport they'd let you lie there for a while. What if you'd done something to your neck and they get you up and put you on your bike and break your neck and you're paralyzed the rest of your life?"

Except in the Benelux countries, helmets are not required for cycling in Europe. Nobody wears them, because they are too hot and, according to most riders, their protective value has not been proven. Race organizers like to point out that competitive cycling is quite safe: "After all," they say, "only six riders have died from injuries in the last 30 years." Of course, this argument ignores the scores of non-fatal injuries sustained in nearly every race: broken legs and arms, dislocated shoulders, amputated fingers, concussions, gashes, and sprains.

Thursday, July 5, saw the first serious individual time trial—67 hilly kilometers from Alençon to Le Mans. Riders were started at two-minute intervals in the inverse order of their overall ranking. On a long time trial, a rider must not think about the distance traveled or the distance to go; rather, he must just focus on the next 20 meters; his attention cannot wander and he cannot relax. His back may be aching, his lungs bursting, and his legs wobbly, but he must pedal on.

Laurent Fignon of the Renault team finished with a time of 1 hour, 27 minutes, 33 seconds, with an average speed of nearly 46 kph. Sean Kelly, an Irish rider, was 16 seconds behind, and Bernard Hinault, an unpopular Breton who had won the Tour four times since his debut in 1978, finished 49 seconds off Fignon's time. Barteau was still in the lead, but Fignon had made up more than five minutes on his total time and was in fourth place overall.

The eighth stage on Friday, July 6 was a largely flat 192-kilometer ride from Le Mans to Nantes at the mouth of the Loire River on the West Coast. Pascal Jules of the Renault team won the stage, although overall he was well down in the rankings. The ninth, another flat stage, saw the cyclists "riding forever," as LeMond put it, "forever" being the 338 kilometers between Nantes and Bordeaux.

At the end of one of the bonus sprints on this long and boring stage, Hinault, accompanied by 20 other riders, made a breakaway and gained 30 seconds on the pack. It didn't take long for the Renault, Panasonic, and Kwantum teams to react. Their riders fell into team time-trial formations and pushed the pace up to 60 kph in an effort to catch the Hinault group. When a team is chasing over a flat road and can see the gap narrowing, the chasers have the psychological advantage. On this particular July day, some additional aid to the chasers came from an unexpected ally, the wind, which shifted from the left side to a headwind. In 20 kilometers the pack had caught the renegades, and things settled down for the remaining 200 odd kilometers. Jan Raas of the Kwantum team was the winner with a time of 9 hours, 40 minutes. The order of the first six riders had not changed since Alençon, and Barteau, surprisingly, was still in first place overall, with the favored Fignon in fourth place, trailing the leader by more than 12 minutes.

The tenth stage—the prelude to the mountains—was held Sunday, July 8. It covered 198 kilometers from Bordeaux to Pau in the foothills of the Pyrenees. Eric Vanderaerden was the stage victor. Fignon displaced Ferreira from third place but was still more than 12 minutes behind race leader Barteau.

Compared to previous years, the climb in the Pyrenees from Pau to the ski resort at the summit of Guzet-Neige (1480 meters high) was relatively benign. Asked why the new route was chosen, the organizers claimed it was to avoid the steeper peaks close to the Spanish border where Spanish Basques, eager for publicity, might possibly disrupt the race. Robert Millar of Scotland was the easy winner with Luis Herrera of Colombia 41 seconds behind him. Overall, the first three places remained unchanged, but Ferreira, who had been holding on to fourth place, proved conclusively that he was not a climber by finishing dead last, 59 minutes and 43 seconds behind Millar.

Next came a series of four stages through the Massif Central, France's mountainous heartland and the transition between the Pyrenees and the Alps. This is a region of hills, gorges, short climbs, and sharp descents. The roads are bumpy and narrow. Stressful climbs, sharp turns, and bold descents took their toll on the riders—both physically and mentally. Grenoble, the start of the tough Alpine stages, was reached on July 14, Bastille Day and the only rest day of the Tour. At this point, after 15 stages, Barteau was still leading with a total elapsed time of 73 hours, 52 minutes, 19 seconds. Fignon had worked up to second place, 10 minutes, 13 seconds behind; Hinault was third, less than two minutes behind Fignon; LeMond was sixth and Kelly seventh, both a bit over 14 minutes behind the leader; and Millar had pulled up to ninth place. Barteau had held up far better than his two companions on the Cergy-Pontoise breakaway; Le Guilloux had dropped to thirty-third while Ferreira, totally defeated by the mountains, was running dead last in the field of 140 men remaining in the race.

The next five days in the Alps would answer many questions:

Could the young Barteau hold onto his lead? Could Hinault win a fifth victory? Could Fignon duplicate his win of the year before? How would the American LeMond make out? Would the Scot, Millar, or the Irishman, Kelly, be any real challenge to the leaders? Would any of the Colombians, known for their mountain racing, be a factor in this race?

First in the Alps was a 22-kilometer individual time trial, followed by daily climbs to L'Alpe-d'Huez, La Plagne, Morzine, and Crans-Montana Mountain in Switzerland. Fignon won the time trial easily; Herrera was second, Delgado third, and Hinault fourth.

The first climb was 151 kilometers; the second, 185 kilometers; the third, 186 kilometers; and the fourth, 140 kilometers. In the first mountain stage, Barteau was unseated from his first-place position. (He continued to slip and finally finished the Tour in 28th place, exactly one hour behind the leader.) By the end of the second of these

mountain stages, it was clear that Fignon would win the race unless he was injured or did something very stupid. Hinault was running second, almost nine minutes behind Fignon, but it was clear that he could not close the gap. LeMond was third, just a little more than one minute behind Hinault, and Millar was just 20 seconds behind him.

As it turned out, these places did not change from the second Alps stage, the eighteenth stage of the race, to the end of the Tour. On the other hand, the mountains exacted a terrible toll from the rest of the pack. Delgado, a fast Spanish rider, got a puncture just before the Morzine finish line, crashed, broke his shoulder, and left the race. Tonon, an experienced Italian rider, hit a spectator who had wandered onto the road with his bike; both were rushed to a hospital by helicopter, Tonon with a fractured skull, the spectator with "grave injuries." In all, 16 riders were forced to retire from the race with injuries during these four stages.

The twenty-first stage was another long one—320 kilometers, from Crans-Montana to Villefranche-sur-Saône, across the Beaujolais wine country. The riders, exhausted from their days in the mountains, appealed to the race organizers to shorten the stage; they won a reduction, and the first 30 kilometers were covered in team cars.

Little was at stake during this stage and during the ten hot and dreary hours, the riders decided to clown a bit. Barteau pretended to read a newspaper, Castaing did a giant crossword puzzle, Brun turned his handlebars up like a casual Sunday rider, and the pack goaded Cabrera, a timid little Colombian, to win two of the bonus sprints. Kelly was not having any of it; while Cabrera was playfully being allowed to win his first sprint, Kelly was right behind him, gaining an eight-second reduction in his overall time. By the end of the day, Kelly had worked his way up from seventh place to fifth. Kelly, desperate for a stage victory, kept pressing on and eventually won a mock victory at the midpoint of the stage.

The next day was a 51-kilometer time trial, and Kelly, determined to earn the coveted stage victory, went all out to win. His time was a sensational 1 hour, 7 minutes, 9 seconds; Hinault was second, 36 seconds behind; and LeMond third, 41 seconds behind. Only Fignon, last to start, was still out on the road. He finished the race with a long sprint in the same time as Kelly; when the timers calculated the time to thousandths of a second, they gave the race to Fignon by 48/1000 seconds. Disappointed and frustrated, Kelly had no use for the condolences of other riders.

Later that day, July 21, the riders boarded a train for Paris. For all intents and purposes, the race was over, although there was one last 196-kilometer stage run around the suburbs of Paris that would pass the Arc de Triomphe six times before it swept down the Champs-Élysées to the finish line at the Place de la Concorde. This was a sprinter's stage, and it was won by Vanderaerden, with Jules, Hoste, Hinault, and Kelly following.

Finishers: 124 out of 170. First: Laurent Fignon. Second: Bernard Hinault. Third: Greg LeMond. Fourth: Robert Millar. Fifth: Sean Kelly. First Colombian finisher: Luis Herrera, twenty-seventh place. Leading climber (polka-dot jersey): Millar. Points leader (green jersey): Franck Hoste. Leading neophyte (white jersey): LeMond. Top team by time: Renault. Top team by points: Panasonic/Raleigh.

Months later, as the season came to a close, Hinault came back to win two prestigious one-day classics; Barteau, being in great demand, rode several six-day races; and Sean Kelly was awarded the Super Prestige Pernod trophy as the season's top-ranked professional bicyclist. LeMond left the Renault team and signed a \$1 million contract with the La Vie Claire team. Three other teams disbanded. And riders mounted their exercise machines to wait for the start of the next season.

BIBLIOGRAPHY

Abt, Samuel. *Breakaway: On the Road with the Tour de France*. New York: Random House, 1985.

Leete, Harley M., ed. *The Best of Bicycling!* New York: Trident Press, 1970.

Walton, Bill and Rostaing, Bjarne. *Bill Walton's Total Book of Bicycling*. New York: Bantam Books, 1985.

THE PROGRAM

The Tour de France program consists of a main program (with five sections and five associated subroutines), two introductory sections (with five associated subroutines), a closing section, and five utility subroutines.

In the introductory section (Lines 100–220), the title screen is displayed followed by a screen of instructions. While this screen is being displayed, data are read in subroutines at Lines 1430, 3050, and 3300. After all the data have been read, you are instructed to "Press any key to continue," and program execution proceeds.

In the pre-race data section (Lines 240–510), you are asked to enter information about your physical fitness, practice time, the amount of pedaling help you want from the computer, and the timing loop value for your computer. This last item is unique to this program and is needed for the pedaling routine.

Tour de France contains a pedaling subroutine (Lines 2890–3030) that simulates in an interesting way the actual pedaling of a bicycle. Specifically, it asks you to alternately press two keys (representing the two pedals on a bike). The program counts the number of keystrokes you make in a given period of time and computes a revolutions per minute (rpm) value. To pedal an actual bike, you must use both feet; so it is with the program:

```
2960 K = 0 : FOR I = 1 TO PT * PTM
2970 A$ = INKEY$ : IF LEN(A$) = 0 OR A$ = B$ THEN 2990
2980 IF A$ = LF$ OR A$ = RT$ THEN K = K + 1 : B$ = A$
2990 NEXT I
```

In the routine, you pedal for a certain amount of time (PT * PTM), which will be discussed later. The INKEY\$ function polls the keyboard. If there has been no keystroke, the length of A\$ is 0 and the routine goes to the next I. Also, if the current keystroke is the same as the previous one (A\$ = B\$)—two pushes with the left foot or two with the right—the routine continues and does not count that keystroke. The routine then tests (Line 2980) whether the keystroke is the same as one of the two pedals (LF\$, RT\$); if so, it is counted as one keystroke (K = K + 1). The value of A\$ is then put into B\$ and it becomes the previous keystroke.

The length of the timing loop (PT * PTM) consists of a fundamental pedaling-time (PT) value for your computer times a pedal-time multiplier (PTM). Variable PT will be different for different processor speeds. For example, a standard IBM PC (or clone) may be able to execute this routine 2000 times in ten seconds, whereas a Tandy 2000 or IBM PC AT may be able to execute it 3600 times. However, it is desirable for you to have the same amount of time to 'pedal' on the keyboard no matter what computer you are using. Hence, PT is a function of your computer and BASIC interpreter. (For a standard PC or clone, the value of PT should be about 2150.) The value of PT need be calculated only once (Lines 2820–2870). It is set equal to twice the number of times the computer executes a rather simple loop in 20 seconds. The real time (in seconds) is equal to:

```
60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2))
```

The real time is always available in the string variable TIME\$. For readability, colons are automatically inserted in this string between the hours, minutes, and seconds, so it returns string values such as 14:35:12 or 10:09:00. The right-hand part of the above expression takes the number of seconds and converts it to a numeric value. The left-hand part takes a string of two numbers starting with the fourth character of the string (minutes) and converts it to a numeric value, which is multiplied by 60 to get the number of seconds.

This loop is run until the current time is equal to the starting time plus 20 seconds. On an IBM PC, the loop is executed about 54 times per second. Thus, running it for a shorter time than 20 seconds leads to somewhat erratic values because the loop may start at any time within the initial second (1/100 to 99/100). Running it for 20 seconds means that the value of PT will be between, say, 2100 and 2200, that is, approximately plus or minus 2.5% of the exact value, a tolerable range for pedaling time.

The pedal-time multiplier is normally equal to one. However, if you elect to do a very long sprint (Lines 1000–1100), PTM could be as high as five, which leads to a very long and exhausting pedaling time.

The routine to select the gear range (Lines 620–830) is straightforward. If you select a gear range that is too high for the terrain, you will have to shift much more than you should, and appropriate penalties are accessed in Lines 820 and 830. In addition, using a gear range that is too high in the mountains increases physical exertion and the risk of collapse. Line 920 tests for these conditions and, if they exist, increases the probability of physical collapse by ten percentage points (from 11% to 21%). If nothing happens, probabilities are returned to normal in Line 940.

In the sprint routine (Lines 1000–1100), a counter, which counts down in 0.1-kilometer increments starting 10 km from the end of the race, is started (Line 1040). Pressing any key interrupts this countdown, and your sprint starts. Note that the countdown values overprint themselves (Line 1050) with the value at which you interrupt the loop being held on the screen.

Daily elapsed times for five other riders are calculated in Lines 1160 to 1180. The program assumes they are all in excellent physical condition and that they use the correct gear for the terrain. They have their share of delay-causing problems (1.4 * RND(1)) which tend to be in the same range as your delay time.

Event probabilities (Lines 1430–1490) are handled in the same generalized way as in the Marco Polo program, the difference being that Tour de France has three different sets of events: road hazards, mechanical breakdowns,

and physical problems. Each of these sets of events has its own subroutine. Various events cause delays of between 0.1 hours (6 minutes) and 1 hour (physical collapse, assuming you recover). A few "events" are simply informative and cause no delay: a nice day in the French countryside, bicycle running like a charm, and feeling fit as a fiddle.

All the other subroutines either are straightforward and self-explanatory or are explained in the program notes for other programs.

PROGRAM VARIABLES

A Answer to input query, numeric
A\$ Answer to input query, string
B\$ Temporary string variable
CG Teeth on cog
CL Physical-collapse indicator
CR Crash indicator
DAY Day of race
DIST(d) Distance to cover (d = day)
DSP Distance of sprint
EN End time for timing loop
FIT Physical-fitness factor (.1 = poor, .5 = excellent)
GDGR Good gear indicator (0 = poor, 1 = good)
GQ Gear ratio x distance (for calculating performance of other riders)
GR Gear ratio
I Temporary iteration variable
K Temporary iteration variable
KPH Speed over route
KSR Speed of sprint
LF\$ Left-pedal key
PAUSE Temporary iteration variable
PD Pedal indicator
PFRQ Pedaling frequency (1 = low, 10 = high)
PLACE\$(d) Destination for day (d = day)
PM(n) Probability of mechanical problem (n = problem no.)
PMT Total probabilities of all mechanical problems
PP(n) Probability of physical problem (n = problem no.)
PPT Total probabilities of all physical problems
PR(n) Probability of road hazard (n = hazard no.)
PRT Total probabilities of all road hazards
PT Pedaling time
PTM Pedaling-time multiplier
RN Random number
RNG Teeth on ring gear
RPM Pedal speed
RPS RPM-speed multiplier
RT\$ Right-pedal key
TDEP Time of departure
TDL Time for delays
TMSD Time to ride sprint distance
TMRD Time to ride route distance
TTM(r) Total time for stage (r = rider)
TTR(r) Total time for race to date (r = rider)
TTS Total stage time, leader
TTT Total race time, leader
TYPE(d) Type of terrain (d = day)
TYPE\$(t) Type of terrain, description
WK Weeks of practice
WS Winner of stage
WSG(r) Stages won by each rider (r = rider)
WT Winner of race to date
X\$ Temporary string variable
Note: Distances are in kilometers
Times in hours and fractions
Speeds in kilometers per hour

PROGRAM LISTINGS

[Download TOUR.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF
110 LOCATE 10, 1 : X$ = "Tour de France Bicycle Race" : GOSUB 3470
120 LOCATE 13, 1 : X$ = "(c) David H. Ahl, 1986" : GOSUB 3470
130 LOCATE 23, 1 : X$ = "Press any key to continue." : GOSUB 3470
140 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND
150 CLS : GOSUB 3500 : 'Display initial scenario
160 DIM PLACE$(23), TYPE(23), DIST(23), PP(14)
170 GDGR = 1 : PLACE$(0) = "Reims"
180 GOSUB 1430 : 'Read event probabilities
190 GOSUB 3050 : GOSUB 3300 : 'Read data about places
200 LOCATE 23, 1 : X$ = "Press any key to continue." : GOSUB 3470
210 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND
220 WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN : CLS
230 '
240 'Enter pre - race data
250 INPUT "Do you have the timing loop value from a previous run";A$
260 GOSUB 3350 : IF A$ = "N" THEN 290
270 INPUT "Please enter the value";PT : IF PT > 500 AND PT < 5000 THEN 300
280 INPUT "That doesn't sound right. Please enter it again";PT : GOTO 300
290 GOSUB 2820 : 'Calculate pedaling time (takes 20 seconds)
300 PRINT : PRINT "About your physical fitness: are you (1) in fantastic health,"
310 PRINT " (2) excellent shape, (3) quite good, (4) okay, (5) poor"
320 INPUT " Please enter a number between 1 and 5";A
330 IF A < 1 OR A > 5 THEN PRINT "Huh? I don't understand." : GOTO 320
340 FIT = .57 - .04 * A : 'Physical fitness factor
350 PRINT "How many weeks do you intend to take off from work or school to"
360 INPUT "practice and prepare for the race";WK : IF WK > 12 THEN WK = 12
370 IF WK > 5 THEN 390 ELSE PRINT "You must be joking. You'll need at least six"
380 PRINT "weeks if you want to be a real contender. Now..." : GOTO 350
390 FIT = FIT - (12 - WK) * .05 : 'Modify fitness factor for amount of practice
400 GOSUB 2760 : 'Assign keys to pedals
410 PRINT "Thank you. Let's go out for a practice run."
420 PFRQ = 10 : PTM = 1 : GOSUB 2890 : PD = 0 : PRINT : 'Pedal bicycle
430 PRINT "In an upper-middle gear ratio (52 / 17)," RPM "rpm would translate"
440 PRINT "to a speed of approximately" INT(.3956 * RPM + .5) "kilometers per hour."
450 IF RPM > 60 THEN 480 ELSE PRINT "That speed is barely competitive. Next time"
460 PRINT "you play, try a timing-loop value of" INT(80 * PT / RPM) "but for now"
470 PRINT "you may want to have the computer do most of the pedaling."
480 PRINT : PRINT "How much do you want to pedal your bike (on the keyboard)?"
490 PRINT "1 = hardly at all, 3 = occasionally, 7 = frequently, 10 = every opportunity"
500 INPUT "Please enter a number between 1 and 10";PFRQ : IF PFRQ < 1 THEN PFRQ = 1
510 IF PFRQ > 10 THEN PFRQ = 10
520 '
530 'Beginning of main riding section
540 DAY = DAY + 1
550 PRINT : PRINT "Date : July" DAY " You are at " PLACE$(DAY - 1) "."
560 IF TYPE(DAY) < 5 THEN 580
570 PRINT "Today, thank goodness, is a rest and recuperation day." : GOTO 540
580 PRINT "Your destination is " PLACE$(DAY) "," DIST(DAY) "km from here."
590 PRINT "Type of racing this stage:" TYPE$(TYPE(DAY))
600 IF TYPE(DAY) < > TYPE(DAY - 1) THEN 650
610 '
620 'Select gear range
630 INPUT "Do you want a different basic gear range than yesterday";A$
640 GOSUB 3350 : IF A$ = "N" THEN 850 ELSE GDGR = 1
650 PRINT "Naturally you will shift gears, but what will be your basic gear"
660 INPUT "range (ring and cog) for the day. First the ring (40 or 52)";RNG
670 IF RNG = 40 OR RNG = 52 THEN 690 ELSE PRINT "You don't have that ring."
680 INPUT "Enter 40 or 52 please";RNG : GOTO 670
690 INPUT "Which cog (13, 15, 17, 19, 21, 23, or 25)";CG
700 IF CG = 13 OR CG = 15 OR CG = 17 OR CG = 19 OR CG = 21 OR CG = 23 OR CG = 25 THEN 720
710 PRINT "Sorry, you don't have that cog. Please try again." : GOTO 690
720 IF (CG = 13 AND RNG = 40) OR (CG = 25 AND RNG = 52) THEN 730 ELSE GR = RNG / CG : GOTO
750
730 PRINT "The chain line will be badly skewed with that combination."
740 PRINT "Let's do it again. First the ring. "; : GOTO 680
750 IF TYPE(DAY) = 4 THEN 850
760 IF GR > 3.2 THEN X$ = "high" ELSE IF GR < 1.8 THEN X$ = "low" ELSE GOTO 790
770 PRINT "That ratio sounds very " X$; : INPUT ". Do you want to change it";A$
780 GOSUB 3350 : IF A$ = "Y" THEN 740
790 IF TYPE(DAY) = 3 AND GR > 2.3 THEN 800 ELSE 830

```

```

800 PRINT "For mountainous terrain, that's a rather high basic gear ratio."
810 INPUT "Do you want to stick with it";A$: GOSUB 3350 : IF A$ = "N" THEN 740
820 GDGR = 1.3 - .19 * GR : GOTO 840 : 'Penalty for too high gear ratio in mts
830 IF GR > 3 THEN GDGR = 1.35 - .14 * GR : 'Penalty for too high gear on flat route
840 '
850 'Start of stage
860 TDEP = 100 + INT(59 * RND(1))
870 PRINT "Your departure time is scheduled at 9 : "; : PRINT RIGHT$(STR$(TDEP), 2)
880 GOSUB 3440 : PTM = 1 : RPS = 130 : GOSUB 2890 : KPH = RPM * .1292706 * GR * GDGR
890 PRINT USING "##.##"; KPH; : PRINT " kph." : PRINT
900 TDL = 0 : GOSUB 1510 : 'Road hazards
910 CR = 0 : PRINT : GOSUB 1760 : 'Mechanical breakdowns
920 IF TYPE(DAY) = 3 AND GR > 2.7 THEN PP(1) = PP(1) + 10 : PPT = PPT + 10 : PPX = 1
930 PRINT : GOSUB 2140 : 'Physical problems
940 IF PPX = 1 THEN PPX = 0 : PP(1) = PP(1) - 10 : PPT = PPT - 10
950 PRINT : PRINT "Time for a quick breather. You have about";
960 PRINT INT(20 + 20 * RND(1)) "km to go."
970 PRINT "Press any key when you're ready to go."
980 WHILE LEN(INKEY$) = 0 : WEND : PRINT "Okay, on the road again..."
990 '
1000 'Sprint at end of stage
1010 GOSUB 3440 : PRINT : PRINT "You're coming up on 10 km from the end."
1020 PRINT "During the countdown (in 0.1-kilometer increments) you can"
1030 PRINT "press any key when you want to start your sprint." : GOSUB 3440
1040 PRINT : FOR DSP = 10 TO 0 STEP -.1 : 'Count down from 10km by 0.1 increments
1050 LOCATE 23, 10 : PRINT USING "##.##";DSP : IF LEN(INKEY$) > 0 THEN 1070
1060 NEXT DSP : KSR = 1 : GOTO 1090
1070 PTM = DSP / 2 : RPS = 140 : GOSUB 2890 : KSR = RPM * .396 : 'Sprint speed
1080 PRINT USING "##.##"; KSR; : PRINT " kph." : PRINT
1090 TMSD = DSP / KSR : TMRD = (DIST(DAY) - DSP) / KPH : TTM(1) = TMSD + TMRD + TDL
1100 IF DSP > 3 THEN PRINT "Puff...puff...puff. That was a L-O-N-G sprint!"
1110 '
1120 'Calculate top six riders
1130 PRINT : PRINT "Race summary (total times in hours) : "
1140 PRINT " Rider 1 (you) 2 3 4 5 6"
1150 TTS = TTM(1) : WS = 1 : PRINT "Stage time", : PRINT USING "#####.##"; TTM(1);
1160 IF TYPE(DAY) = 3 THEN GQ = .3 ELSE GQ = .4
1170 FOR I = 2 TO 6 : RPM = 70 + 20 * RND(1)
1180 TTM(I) = DIST(DAY) / (GQ * RPM) + 1.4 * RND(1) : PRINT USING "#####.##"; TTM(I);
1190 IF TTM(I) < TTS THEN TTS = TTM(I) : WS = I
1200 NEXT I
1210 TTT = 1000 : WT = 0 : PRINT : PRINT "Total time", : FOR I = 1 TO 6
1220 TTR(I) = TTR(I) + TTM(I) : PRINT USING "#####.##"; TTR(I);
1230 IF TTR(I) < TTT THEN TTT = TTR(I) : WT = I
1240 NEXT I : PRINT : PRINT
1250 PRINT " Stage winner : Rider" WS " Overall leader : Rider" WT
1260 WSG(WS) = WSG(WS) + 1
1270 IF DAY < 22 THEN 540
1280 '
1290 'End-of-race summary
1300 PRINT : PRINT "The Tour de France has ended!" : PRINT : X = 0
1310 FOR I = 1 TO 6 : IF WSG(I) > X THEN X = WSG(I) : WS = I
1320 NEXT I : PRINT "Winner of the most stages (" X ") was Rider" WS;
1330 IF WS = 1 THEN PRINT " That's YOU!" ELSE PRINT
1340 PRINT "Overall winner by elapsed time was Rider" WT;
1350 IF WT = 1 THEN PRINT " That's YOU!" ELSE PRINT
1360 TTT = 1000 : WT = 0 : FOR I = 1 TO 6
1370 IF TTR(I) - 2 * WSG(I) < TTT THEN TTT = TTR(I) - 2 * WSG(I) : WT = I
1380 NEXT I
1390 PRINT "Overall points winner (time and stages) was Rider" WT;
1400 IF WT = 1 THEN PRINT " That's YOU!" ELSE PRINT
1410 GOTO 2730
1420 '
1430 'Subroutine to read event probabilities
1440 FOR I = 1 TO 10 : READ A : PRT = PRT + A : PR(I) = PRT : NEXT I
1450 DATA 5, 5, 5, 5, 5, 5, 5, 5, 5, 5
1460 FOR I = 1 TO 8 : READ A : PMT = PMT + A : PM(I) = PMT : NEXT I
1470 DATA 5, 5, 5, 5, 5, 5, 10, 5
1480 FOR I = 1 TO 14 : READ A : PPT = PPT + A : PP(I) = PPT : NEXT I : RETURN
1490 DATA 8, 5, 5, 5, 5, 5, 5, 5, 5, 8, 5, 5, 5, 3
1500 '

```

```

1510 'Subroutine to deal with road hazards
1520 RN = INT(PRT * RND(1)) : FOR I = 1 TO 10
1530 IF RN > PR(I) THEN NEXT I : I = 10 : 'If event happened, exit loop
1540 ON I GOTO 1550, 1570, 1590, 1610, 1630, 1650, 1670, 1690, 1720, 1740
1550 PRINT "Mostly gravel roads this stage. You'll have to slow down."
1560 TDL = TDL + .8 : RETURN
1570 PRINT "The roads in this area are very bumpy and will slow you down."
1580 TDL = TDL + .5 : RETURN
1590 PRINT "Hot weather in this area has caused the roads to become very"
1600 PRINT "slippery from oil seepage." : TDL = TDL + .3 : RETURN
1610 PRINT "The wind is at your back making for a very fast ride!"
1620 TDL = TDL + -.3 : RETURN
1630 PRINT "You're heading straight into the wind today. Tough going."
1640 TDL = TDL + .5 : RETURN
1650 PRINT "There is a gusty sidewind today creating balance problems."
1660 TDL = TDL + .3 : RETURN
1670 PRINT "Dreary day: drizzle, fog, and clammy chill in the air."
1680 TDL = TDL + .2 : RETURN
1690 PRINT "Horrible weather! Icy rain that hits you like 1000 needles,"
1700 PRINT "stinging your face and arms. Your shoes are soaked. And there"
1710 PRINT "are few spectators to cheer you on." : TDL = TDL + .5 : RETURN
1720 PRINT "Mud and puddles on the road cause you to slide and skid all over."
1730 TDL = TDL + .4 : RETURN
1740 PRINT "Today is a crisp, clear day in the French countryside." : RETURN
1750 '
1760 'Subroutine to deal with mechanical breakdowns
1770 RN = INT(PMT * RND(1)) : FOR I = 1 TO 8 : 'Select event
1780 IF RN > PM(I) THEN NEXT I : I = 8 : 'If event happened, exit loop
1790 ON I GOTO 1800, 1830, 1850, 1890, 1930, 1970, 1990, 2120
1800 INPUT "You have a broken spoke. Want to fix it now";A$ : GOSUB 3350
1810 IF A$ = "Y" THEN TDL = TDL + .1 ELSE TDL = TDL + .15
1820 RETURN
1830 PRINT "You got a flat tire. You'll have to change it now."
1840 TDL = TDL + .1 : RETURN
1850 PRINT "Your brakes tend to lock every time you apply them hard. You can"
1860 INPUT "nurse them along or fix them here. Want to fix them now";A$
1870 GOSUB 3350 : IF A$ = "Y" THEN TDL = TDL + .2 ELSE TDL = TDL + .4
1880 RETURN
1890 PRINT "You seem to be missing shifts to your 19 cog. Perhaps one or"
1900 PRINT "more teeth are worn. You can shift around it or fix it here."
1910 INPUT "Want to fix it now";A$ : GOSUB 3350 : IF A$ = "Y" THEN TDL = TDL + .2 : RETURN
1920 TDL = TDL + .4 : RETURN
1930 PRINT "On a tight corner, you narrowly missed a spill, but your toe clip"
1940 INPUT "got bent on a boulder near the road. Want to bend it out now";A$
1950 IF A$ = "Y" THEN TDL = TDL + .1 ELSE TDL = TDL + .2
1960 RETURN
1970 PRINT "Uh oh! Chain broke. You've no choice but to fix it now."
1980 TDL = TDL + .15 : RETURN
1990 PRINT "WHOOPS! Took a corner too fast, lost traction, slid, and CRASHED!"
2000 GOSUB 3440 : CR = 1 : RN = RND(1) : IF RN < .03 THEN 2080 ELSE IF RN < .5 THEN 2040
2010 PRINT "You pick up yourself and your bicycle. You're both"
2020 PRINT "scratched and a bit beaten up but there seems to be no"
2030 PRINT "serious damage so you get on your way." : TDL = TDL + .3 : RETURN
2040 PRINT "You twisted your ankle and it is very painful. You know it will"
2050 PRINT "slow you down. However, there is no way you would drop out of the"
2060 PRINT "race, so you pick up your bicycle and get on your way."
2070 TDL = TDL + .8 : RETURN
2080 PRINT "Blood is all over the place; ambulance is called and"
2090 PRINT "you are rushed to the local hospital." : GOSUB 3440
2100 PRINT "Bad news! You dislocated your shoulder and you're out of the race."
2110 GOTO 2700
2120 PRINT "Bicycle ran like a charm today. No problems at all!" : RETURN
2130 '
2140 'Subroutine to deal with physical problems
2150 RN = INT(PPT * RND(1)) : FOR I = 1 TO 14
2160 IF RN > PP(I) THEN NEXT I : I = 14 : 'If event happened, exit loop
2170 IF I > 10 THEN 2190
2180 ON I GOTO 2200, 2320, 2340, 2370, 2390, 2420, 2440, 2470, 2500, 2530
2190 ON I - 10 GOTO 2560, 2600, 2640, 2660
2200 X = INT(DIST(DAY) / 50) : IF X < 2 THEN X = 2
2210 PRINT "You're pushing yourself to the absolute limit and after" X "hours"

```

```

2220 PRINT "you totally collapse. The medics give you oxygen and bring you"
2230 PRINT "around, but warn you against resuming the race." : IF CL > 0 THEN 2290
2240 CL = 1 : IF RND(1) > .8 THEN 2270
2250 GOSUB 3440 : PRINT "But nothing can defeat your competitive spirit and you"
2260 PRINT "vow to press on regardless." : TDL = TDL + 1 : RETURN
2270 GOSUB 3440 : PRINT "You heard of another rider dying from overexertion last"
2280 PRINT "year, so you follow the doctor's advice and withdraw." : GOTO 2700
2290 GOSUB 3440 : PRINT "This is the second time you collapsed in this race,"
2300 PRINT "so you reluctantly concede that this just isn't your year"
2310 PRINT "and you withdraw from the race." : GOTO 2700
2320 PRINT "You have a terrible abdominal pain...something you ate, perhaps?"
2330 PRINT "You'll have to slow down a bit." : TDL = TDL + .4 : RETURN
2340 PRINT "You're having difficulty breathing and you're feeling lightheaded."
2350 PRINT "You recognize this as an early warning signal of total collapse"
2360 PRINT "and wisely decide to slow your pace a bit." : TDL = TDL + .3 : RETURN
2370 PRINT "You seem to be seeing through a haze...and it's not the weather."
2380 PRINT "Occasionally, you can't seem to focus at all." : GOTO 2350
2390 PRINT "Uh oh! A muscle in your calf seems to have turned to jelly. It's"
2400 PRINT "not particularly painful, but it seems to be completely out of"
2410 PRINT "control. You'll have to slow down a bit." : TDL = TDL + .3 : RETURN
2420 PRINT "You have a sharp pain in your lower back. It doesn't seem to be"
2430 PRINT "injured...perhaps you're just overly tense." : TDL = TDL + .2 : RETURN
2440 PRINT "The gearing you've been using is really tough on your legs and"
2450 PRINT "you have developed shin splints. You'll have to back off your"
2460 PRINT "blistering pace a bit." : TDL = TDL + .3 : RETURN
2470 PRINT "Terrible pain in the balls of your feet. Your toe clip seems to be"
2480 PRINT "adjusted correctly. Maybe it's these new cleats. In any event,"
2490 PRINT "you decide to back off a bit...just for today." : TDL = TDL + .3 : RETURN
2500 PRINT "A medic takes a look at you during lunch break and declares you"
2510 PRINT "have a salt/water imbalance. 'Drink more water along the way,'"
2520 PRINT "he recommends, 'and don't forget your salt pills.'" : RETURN
2530 IF TYPE< >3 THEN 2660
2540 PRINT "The altitude is getting to you in the mountains. You're short of"
2550 PRINT "breath and you feel lightheaded. : goto 2445
2560 PRINT "Your saddle feels like it has appended itself to your body. A cyst"
2570 PRINT "seems to be starting, something you want to avoid at all costs."
2580 PRINT "You put some extra padding on the saddle and back off on your"
2590 PRINT "pace just a tad." : TDL = TDL + .15 : RETURN
2600 PRINT "The blistering pace you've been keeping has played havoc with your"
2610 PRINT "knees. You've heard of football players with bad knees, but a"
2620 PRINT "bicycle racer? Nevertheless, you'll have to slow down a bit."
2630 TDL = TDL + .2 : RETURN
2640 PRINT "You developed a bad cramp in your legs. You'll have to take it"
2650 PRINT "just a bit easier." : TDL = TDL + .15 : RETURN
2660 IF CR = 1 THEN 2150 : 'Can't feel too great after a crash
2670 PRINT "You're feeling fit as a fiddle and have no physical problems today."
2680 RETURN
2690 '
2700 'It's all over
2710 PRINT : PRINT "Too bad. That's it for this year, but there's always"
2720 PRINT "next year...."
2730 PRINT : INPUT "Would you like to ride again";A$ : GOSUB 3350
2740 IF A$ = "Y" THEN RUN ELSE KEY ON : CLS : PRINT "Bye for now." : END
2750 '
2760 'Subroutine to assign keys to pedals
2770 PRINT "To pedal your computer bike, you will strike two keys alternately"
2780 PRINT "With two fingers (one hand or two, it's up to you)."
```

2790 INPUT "Which key do you want for your left pedal";LF\$

2800 INPUT "and which key for the right";RT\$: RETURN

2810 '

2820 'Subroutine to calculate time to pedal as a function of processor speed

2830 PRINT "Calculating timing-loop value takes 20 seconds. Please be patient."

2840 PRINT : EN = 20 + 60 * VAL(MID\$(TIME\$, 4, 2)) + VAL(RIGHT\$(TIME\$, 2))

2850 WHILE 60 * VAL(MID\$(TIME\$, 4, 2)) + VAL(RIGHT\$(TIME\$, 2)) < EN : PT = PT + 1 : WEND

2860 PT = PT * 2 : PRINT "The timing-loop value for your computer is"PT

2870 PRINT "Please write it down for playing this game in the future." : RETURN

2880 '

2890 'Subroutine to pedal bicycle

2900 IF DAY = 8 AND PD = 0 THEN 2950 : 'Must get at least one chance to pedal

2910 IF DAY = 17 AND PD = 1 THEN 2950 : 'Second chance to pedal

2920 IF PFRQ >= 9.92 * RND(1) THEN PD = PD + 1 : GOTO 2950 : 'Human pedal bike?

```

2930 RPM = INT((RPS + 40 * RND(1)) * FIT) : PRINT "Computer is pedaling your bicycle."
2940 GOSUB 3440 : X$ = "It" : GOTO 3020
2950 PRINT "Start pedaling"; : GOSUB 3440 : PRINT " NOW!" : PRINT
2960 K = 0 : FOR I = 1 TO PT * PTM
2970 A$ = INKEY$ : IF LEN(A$) = 0 OR A$ = B$ THEN 2990
2980 IF A$ = LF$ OR A$ = RT$ THEN K = K + 1 : B$ = A$
2990 NEXT I
3000 PRINT "Okay, stop pedaling." : RPM = INT(.9 * FIT * K / PTM) : X$ = "You"
3010 IF RPM > 95 THEN RPM = INT(84 + 10 * RND(1))
3020 PRINT X$ " pedaled at a rate of" RPM "rpm. Calculating speed....";
3030 A$ = "" : FOR I = 1 TO 800 : B$ = INKEY$ : NEXT I : RETURN
3040 '
3050 'Subroutine to read data about routes
3060 FOR I = 1 TO 22 : READ N, PLACE$(I), TYPE(I), DIST(I) : NEXT I : RETURN
3070 DATA 1, "Lille", 1, 213
3080 DATA 2, "Arras", 4, 52
3090 DATA 3, "Caen", 1, 308
3100 DATA 4, "Le Mans", 1, 172
3110 DATA 5, "Nantes", 1, 192
3120 DATA 6, "Bordeaux", 1, 338
3130 DATA 7, "Biarritz/Bayonne", 1, 184
3140 DATA 8, " Lourdes", 3, 168
3150 DATA 9, "Lourdes", 5, 0
3160 DATA 10, "Toulouse", 2, 172
3170 DATA 11, "Rodez", 2, 176
3180 DATA 12, "Avignon", 2, 294
3190 DATA 13, "Grenoble", 2, 228
3200 DATA 14, "l'Alpe-d'Huez", 4, 39
3210 DATA 15, "Lansleburg", 3, 225
3220 DATA 16, "Martigny, Switzerland", 3, 230
3230 DATA 17, "Annecy", 3, 218
3240 DATA 18, "Annecy", 5, 0
3250 DATA 19, "Lyon", 2, 182
3260 DATA 20, "Dijon", 1, 212
3270 DATA 21, "Fountainbleau", 1, 259
3280 DATA 22, "Paris", 1, 210
3290 '
3300 'Subroutine to read words
3310 FOR I = 1 TO 5 : READ TYPE$(I) : NEXT I : RETURN
3320 DATA "Mostly flat with small hills.", "Hills, gorges, steep slopes."
3330 DATA "Mountains." ,"Time trial against the clock.", "Rest."
3340 '
3350 'Subroutine to read yes/no answer
3360 GOSUB 3390 : IF A$ = "Y" OR A$ = "N" THEN RETURN
3370 INPUT "Don't understand your answer. Enter 'Y' or 'N' please";A$ : GOTO 3360
3380 '
3390 'Subroutine to read first letter of answer and convert to uppercase
3400 IF A$ = "" THEN A$ = "Y" : RETURN
3410 A$ = LEFT$(A$, 1) : IF A$ >= "A" AND A$ <= "Z" THEN RETURN
3420 A$ = CHR$(ASC(A$) - 32) : RETURN
3430 '
3440 'Subroutine to make a short pause
3450 FOR PAUSE = 1 TO 300 : X = X + 1 : NEXT PAUSE : RETURN
3460 '
3470 'Subroutine to print a centered line
3480 PRINT TAB((70 - LEN(X$)) / 2) X$ : RETURN
3490 '
3500 'Subroutine to display initial scenario
3510 X$ = "Tour de France Bicycle Race" : GOSUB 3470 : PRINT
3520 PRINT " You are a bicycle racer entered in the 22-day Tour de France"
3530 PRINT "bicycle race around France. Your objective is to win the race by"
3540 PRINT "having the lowest overall elapsed time. In addition, you must try"
3550 PRINT "to win as many individual 'stages' or daily races as possible, as"
3560 PRINT "wins on these stages count toward the overall points prize."
3570 PRINT " Each day you pedal your bicycle by alternately pressing two"
3580 PRINT "keys on your computer keyboard as quickly as possible (you may opt"
3590 PRINT "to have the computer do some of the pedaling for you)."
3600 PRINT " While racing, various hazards occur (weather, mechanical"
3610 PRINT "breakdowns, road conditions, and physical problems) that hamper"
3620 PRINT "your progress."
3630 PRINT " At the end of each stage (day), you may sprint to the finish"

```



```
3640 PRINT "line. The computer will count down the distance starting ten"
3650 PRINT "kilometers from the end of the race. During this countdown, you"
3660 PRINT "must decide when to start your sprint. Remember, if you start too"
3670 PRINT "soon, you may become too exhausted to maintain your sprint to the"
3680 PRINT "end, but if you start too late, other riders may overtake you."
3690 RETURN
```

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



THE GAME

Appalachian Trail places you in the boots of a hiker whose goal is to walk the entire length of the 2000-mile Trail from Springer Mountain, Georgia, to Mt. Katahdin, Maine. Your journey begins in April as soon as the southern part of the Trail is clear of snow, and you must reach the northern terminus before the early snowfalls block the Presidential Mountain Range in New Hampshire.

Along the way, you encounter the same hazards reported in the diaries of hundreds of hikers. Some are natural hazards—prolonged heavy rain, late snowstorms, poison ivy; others are physical problems—hypothermia, excessive weight loss, blisters; and still other problems involve difficulties with equipment—torn raingear, worn-out boots, and broken zippers. It is not easy to complete the Appalachian Trail, but your chances of success will be better if you remember the following:

- Careful planning is important. When deciding what to pack, you must make many trade-offs—generally among comfort, weight, and food.
- Your method of food supply has a great effect on the success of your hike. You must choose one of the following methods:
 - A. Bury caches of food along the trail. This has to be done prior to your hike and requires a great deal of planning.
 - B. Purchase food as you go along. This is the cheapest method, but requires time-consuming off-trail excursions.
 - C. Have a friend send food to post offices along the way. This is one compromise that works well for many hikers, but this method also requires a great deal of planning.

D. The nutritional value of your food is important. You must decide how many calories you want to replenish each day and how to get them from each of the five food groups.

Prepare carefully and deal intelligently with the obstacles you encounter, and five million steps later you can join the elite group of fewer than 2000 hikers who have completed the Appalachian Trail. Happy hiking!



THE STORY

The Appalachian Trail is a continuous footpath that roughly follows the crest of the Appalachian mountain range through 14 states from Springer Mountain in northern Georgia to Mt. Katahdin in the central Maine wilderness.

In 1922, the first mile of the trail was cut and marked in Palisades Interstate Park in New York. Fifteen years later, in August 1937, the trail was proclaimed complete when the last two stretches—the south slope of Mt. Sugarloaf in Maine and a dense section near the Davenport Gap in Tennessee—were finally cut and marked. Over the years many sections have been relocated, particularly in the Great Smoky Mountains and Shenandoah National Park, but the total length of the trail remains a bit over 2000 miles.

The Appalachian Trail Conference, organized in 1925 when the project was barely under way, coordinated the effort to build what is today the longest marked footpath in the world. Today the Conference continues to coordinate the efforts of the organizations and individuals who maintain and preserve the Trail. Although the U.S. Department of the Interior is responsible for the administration of the Trail, volunteers still play the major role in clearing, maintaining, and "operating" the Trail.

The Appalachian Mountain system is not an unbroken range, but consists of many mountain ranges often separated by deep valleys. Some of the more notable ranges include the Great Smokies, the Blue Ridge Mountains, the Alleghenies, the Catskills, the Green Mountains, and the White Mountains.

The Trail passes through 14 eastern states. West Virginia has the least Trail mileage of any state—5.2 miles with another 20.0 miles along its border with Virginia—while Virginia boasts the longest section of trail—464.7 miles. The most mountainous stretches are in Vermont and New Hampshire. The roughest stretches are in Pennsylvania and New Jersey, and the worst weather is generally thought to occur in Maine.

Myron H. Avery was the first person to walk the entire length of the Appalachian Trail, starting in Maine in the 1920s when the Trail was still in its formative stage and finishing the last section 14 years later in the fall of 1936. Another early hiker who walked the entire trail, again in several sections, was George W. Outer-bridge, who started in 1932 and finished in 1939.

When in 1948 Earl Shaffer set out to walk the entire Trail in one continuous trip, most people thought the task quite impossible and him foolhardy to try it. Until that time, the Trail had been in existence for some 11 years and only six or seven people had walked the entire length, each doing so in a series of relatively short hikes over a period of years.

Shaffer had served 41/2 years in the Army in World War II, most of it in the Pacific theater, and had a yearning to spend some time alone in the wilderness. He prepared himself by doing jujitsu exercises and abstaining from smoking. He carried what was considered at the time a light load in a rucksack and set out from Mt. Oglethorpe in Georgia on April 4, 1948. He carried mostly dried foods augmented by food purchased along the route.

Shaffer had ordered a complete set of guidebooks and maps, but they did not arrive in time for his trip. Thus, he relied upon the trail markers, often quite poor, his compass, and instinct to find his way. The trail was in bad condition because maintenance had been curtailed during wartime; indeed, entire sections were obliterated by nettles, poison ivy, and briars, and logging had destroyed many blazed trees and markers.

Nevertheless, with a great deal of perseverance Shaffer managed to reach Mt. Katahdin in Maine on August 5, 1948, just 124 days after setting out. He averaged 16.5 miles per day, but because of detours and leaving the trail for food and supplies, his actual average daily mileage was probably closer to 18 or 19. This is a remarkable average, even for hikers today carrying lightweight equipment on a well-marked trail.

Seventeen years later, in 1965, when he was 45, Shaffer hiked the Trail again, this time from Mt. Katahdin to Springer Mountain. He thus became the first person both to hike the Trail continuously and to hike it continuously in both directions. On his second hike, he left on July 19, 1965 (after black-fly season in Maine) and reached

Springer Mountain in freezing weather on October 25, exactly 99 days later. His average daily mileage was a spectacular 20.5, a rate few hikers have exceeded since.

Another early through hiker was Gene Espy, who, at age 24 in 1951, hiked from Mt. Oglethorpe, Georgia, to Mt. Katahdin in exactly four months. Unlike some other hikers, he prepared no food caches and made no arrangements to be met anywhere along the way. Like most early hikers, Espy carried a fairly heavy load—45 pounds—even though he had only minimal extra clothing and equipment.

Espy is one of the few hikers who report having seen more than one or two rattlesnakes. In his diary he records killing nearly 20, the largest nearly 4 feet long. He also notes seeing four copperheads within a 20-minute period along one section of the Trail in Pennsylvania. These snake encounters made Espy a staunch proponent of carrying a strong walking stick.

Another notable early through hiker was Mrs. Emma Gatewood, better known as Grandma Gatewood, who, in the summer of 1955 at age 67, walked the entire Trail, thus becoming the sixth person and the first woman to complete it. Seventeen months later, in April 1957, she was back on the Trail again, completing it in September. Her first journey took 146 days, and her second took 142 days. Her pack of less than 20 pounds was probably the lightest ever taken on the Trail, and she rarely cooked a meal. Unlike many other hikers who chose heavy military boots, Grandma Gatewood wore lightweight men's Keds, wearing out six pairs in the course of her journey. Neither did she carry a tent, sleeping bag, nor regular backpack; instead she carried a blanket and a rain cape in a homemade shoulder bag. Also, contrary to the rules of hiking, she carried no guidebook, map, or compass.

Over the next seven years, Grandma Gatewood continued to hike various sections of the Trail, some in a north-south direction and, in 1964, when she was age 77, these sections totaled a third completion of the Trail. Moreover, she took time out in 1959 to walk the entire Oregon Trail in honor of its 100th anniversary.

The only other person to have completed the trail three times was also a woman, Dorothy Laker. In contrast to Grandma Gatewood's style, Dorothy Laker did a thorough job of planning for her hikes and carried the best equipment she could afford. It paid off, and she completed her first trip in 1957 in 161 days. Her second trip, which started on June 14, 1962, had to be aborted on October 17 at Mt. Moriah, New Hampshire. On her third hike she got an earlier start, May 4, 1964, and successfully completed the Trail 151 days later on October 2.

Nine years later, in 1971, Laker attempted to complete the last section of her second hike, setting out from Mt. Moriah on August 23, 1971. Unfortunately, she twisted her ankle badly in a boggy area near Saddleback and shortly thereafter a stretch of particularly cold, wet weather set in. She pushed on, but she fell while fording the swollen Kennebec River, ending her attempt to finish the Trail that year. Finally, the next year, more determined than ever, she finished the last 152 miles in a grueling 12-day effort.

There are nearly as many approaches to hiking the Trail as there are people who attempt to do it. Let's consider the various pros and cons of different approaches.

Shelter: There are campsites along the Trail, in most cases no more than a day's hike apart. At many of these campsites there are roofed shelters, generally three-sided with an open front. Shelter facilities are, according to the Appalachian Trail Conference, "primarily provided for the long-distance hiker who may have no other means of shelter. Persons planning short hikes are asked to consider this and plan to carry tents." All very nice in theory, but scores of hikers note that all too often the shelters are packed with Boy Scouts, day campers, or fishermen who have made a shelter their base of operations.

The obvious alternative to using trail shelters is to carry a tent, although many hikers in the fifties and sixties reported great satisfaction in discarding a heavy tent along the way by cutting it up to use as a groundcloth. However, tent design has made great strides since then, and lightweight, modestly roomy tents are now widely available. Needless to say, the larger the tent, the more it weighs, although there is something to be said for a tent large enough to accommodate both you and your pack.

Sleeping: Grandma Gatewood and some other early hikers swore by blankets, but again, the latest lightweight sleeping bags weigh very little more than a single wool blanket yet are considerably more versatile. The key questions today usually boil down to bag shape (mummy—very confining and very warm; rectangular—roomy but heavy; and semi-rectangular—a cross between mummy and rectangular) and insulation material (goose down—warm, lightweight, and expensive—or a synthetic material—heavier than down but cheaper and much faster to dry when it gets wet).

Many of the shelters have bedframes with cut and broken springs. These are murder on sleeping bags, not to mention uncomfortable. Hikers are well advised to carry a heavy groundcloth to protect sleeping bags from these

hazards, as well as for those nights when they must sleep outside on the ground. The floors of shelters are often made from rough logs that are equally uncomfortable; hence, an air mattress or foam pad is a practical necessity.

Cooking: For the most part, the campfire is an item of nostalgia due to the scarcity of dry wood and the danger of forest fires.

Indeed, open fires are not permitted anywhere along the Trail except in fireplaces provided at designated campsites. Therefore, most people carry a small stove. The technology of stoves has also improved greatly during the last decade, and extremely lightweight units which burn a variety of fuels (white gas, propane, and butane) are now available. Some of these fuels, especially white gas, can be purchased easily along the Trail, while others can be carried in compact cylinders or fuel bottles.

Of course, some hikers dispense with cooking altogether and eat cold food on the Trail, stopping for an occasional hot meal at a restaurant or store along the way.

Food supply: The three common methods of obtaining food are buried caches, packages sent ahead and held at post offices, and purchasing needed supplies at stores along the route. Each has its advantages and disadvantages. Food caches require no time-consuming off-trail excursions. Furthermore, canned goods can be buried along with the usual dried and concentrated foodstuffs, thereby giving the hiker a substantial dinner and breakfast when he digs up a cache. Depending upon how many caches are buried—some hikers bury as many as 50—the food load carried by the hiker can be potentially very light.

Purchasing food is the cheapest method of supply, because stores stock ordinary groceries rather than expensive trail mixes. On the other hand, this method requires many off-trail excursions, some of which can be quite long, and the food carried can be relatively heavy.

Food sent to and held at post offices along the way is a compromise; the hiker can send lightweight camp-supply meals and augment them with food purchased from stores, usually found near the post office.

No matter what the method of resupply, most hikers' diets are unbalanced and quite different from what they normally eat. Almost invariably the trail diet is deficient in fruits and fresh vegetables and abundant in grains and fibers. Because he is walking all day, the hiker burns up far more calories than he would in the course of a normal day at home. All these factors mean that he is calling upon his body to make some fairly radical adjustments in a short period of time. Some people's bodies adjust better than others', but nearly everyone experiences diet-related problems to some extent.

Water: It is advisable to carry a canteen or water bag to assure a supply of potable water. In many areas along the Trail the water is unsafe for drinking, and a considerable distance may pass before potable water can be found. All water obtained from unprotected or open sources should be purified by boiling or by chemical treatment before use.

Pack: Hikers constantly debate the relative merits of the external versus internal frame pack. The external frame pack rides farther away from the back and is cooler on hot days. Sleeping bag and mattress are usually carried on the outside and can be attached to the pack frame in a way that gives optimum weight distribution. Because these items are carried on the outside, an external frame pack generally holds more, but it also weighs more.

Everything normally is put inside of an internal frame pack, although some hikers still tie a sleeping pad to the outside. Proponents say that an internal frame pack keeps a hiker's things drier and is lighter and more compact than an external frame pack.

Footwear: Boots, say some hikers, make or break the trip.

Grandma Gatewood wouldn't agree, but in this she is clearly in the minority. Probably the major debate about footwear these days is leather versus synthetics. Many hikers still swear by a heavy handcrafted pair of one-piece leather boots, while others like a synthetic blend, alone or in some combination with leather, for its ability to breathe and dry quickly.

It is rare that a pair of boots, even the best custom-made pair, will last the entire trail without needing at least new soles. New soles usually last about 800 to 1000 miles; the rocky areas of Pennsylvania and New Jersey will do in a second pair; and the mountains of New England will finish off a third.

Raingear: When it rains on the Appalachian Trail it often comes down for days or weeks at a time. Thus, no matter what kind of raingear he carries, the hiker gets wet as the water seeps under his clothes, down his neck, and into his boots. Therefore, one approach is to not carry any raingear at all and to dry off when stopping at night but not

otherwise. Most people still attempt to keep partially dry and carry a poncho or rain parka, and some hikers even carry rain pants.

Hazards: Most through hikers report that the first thing non-hikers ask about is snakes. In fact, snakes avoid human contact and many hikers have walked the entire Trail without seeing a single one. Nevertheless, it is probably better to err on the side of caution, and practically every hiker carries a 1 oz. snakebite kit.

For the most part, other animals also avoid humans, except if they have been fed by people (as have some of the bears in the Great Smoky Mountain National Park) or if they have been successful in the past finding food open in packs (raccoons, porcupines, and skunks usually do). Mice, which make their homes in many of the shelters, are always a problem.

But by far the most dangerous animals are the dogs found roaming about the small towns from Georgia to Pennsylvania. Not afraid of humans, they are perhaps the worst enemies hikers face on the trip. Against curs, as well as many other animals, the best defense is to avoid them and the second defense is a heavy hiking stick.

For many hikers, the worst hazard along the Trail is the poison ivy, oak, and sumac. In some stretches it is impossible to go around the thickets of poison ivy, and the hiker who has any susceptibility at all will probably come down with a case.

Probably the most common injury is a twisted ankle. The many rock scrambles, stream crossings, and boggy areas offer ample opportunity to put a foot down on an unstable surface.

A long hike is rough on equipment, and most hikers rip, tear, and wear out socks, pants, shirts, zippers, straps, stuff sacks, tents, groundcloths, and practically everything else at an amazing rate. It is vital to carry a small sewing kit and make repairs at the first sign of a rip. Worn socks should be replaced immediately to avoid blisters, whereas other items can be replaced later.

Trail markings: In theory, from each blaze on the Trail it is possible to see the next one. In practice, it isn't so. The Trail often has long sections where there are few opportunities for orientation or checking the route. The Trail guidebooks advise hikers never to proceed more than a quarter mile (roughly five minutes of hiking) without noticing some Trail indication. The cardinal mistake is an insistence on going forward when the route seems obscure or dubious.

Major stretches of the Trail are relocated almost every year because of commercial or residential development. Thus, even the latest maps and guidebooks are invariably out of date. The hiker must remain alert to the painted blazes and, if they deviate from the guidebook, should follow the blazes in preference to the printed data. In practice, this means that he will almost inevitably get lost—or at least stray from the Trail—more than once during the trip.

A decision to walk the Trail is not something to be undertaken lightly. A major physical and mental commitment is needed—a commitment to being alone and independent for three to seven months; a commitment to overcoming boredom, discomfort, weather, and hazards; a commitment to forcing the body to do more than it has ever done before; and a commitment to enjoying the challenge and adventure of a lifetime.

BIBLIOGRAPHY

Appalachian Trail Guides. *Appalachian Trail Conference*. Harpers Ferry, WV, 1984.

Fletcher, Colin. *The New Complete Walker*. New York: Knopf. 1982.

Hare, James R. (ed). *Hiking the Appalachian Trail* (2 vols). Emmaus, PA: Rodale Press, 1975.

Look, Dennis. *Joy of Backpacking*. Sacramento, CA: Jalmar Press, 1976.

Manning, Harvey. *Backpacking: One Step At a Time*. Seattle, WA: REI Press, 1972.

THE PROGRAM

At over 550 lines, the Appalachian Trail program is by far the largest in this book. There are two reasons for this: First, such a trip requires a tremendous amount of personal preparation—food, clothing, equipment, physical conditioning—which, in the program, requires over 175 lines of code. Second, walking the Trail is a complex undertaking, requiring many decisions and trade-offs. For example, you may set a brisk hiking rate for yourself.

However, this means you will burn more calories and have to eat (and carry) more food. In turn, this means that your load is heavier, which slows you down and causes you to burn even more calories.

Not all of the possible interactions have been included in the program and you may wish to modify it to include others. Some suggestions for doing this are included in the description of the program.

Because this program uses many of the same techniques and routines as other programs in this book, only those that are unique or suitable for modification are discussed.

You are asked to make decisions about seven major items of equipment: tent, pack, sleeping bag, foam pad, stove, boots, and raingear (Lines 550–910). If you opt for an internal frame pack, everything you take must fit inside (Line 680), whereas with an external frame pack, your sleeping bag and foam pad may be carried on the outside (Line 670). I have assumed a volume of 3000 cu. in. for clothing, food, toilet supplies, eating utensils, guidebooks, etc. If you feel this is too little or too much, the volume can be changed. Also, the descriptions of the main items can be changed in the data statements (Lines 4740–5080). Each item requires four pieces of data: name, weight (ounces), price (dollars), and volume (cu. in.).

Actually, prices are not used, except to give you some idea of the cost of everything if purchased new. However, as many hikers seem to stay on rather limited budgets, it would be easy to add a routine to make sure that you stay within a certain total cost.

With respect to the food to be carried, you are asked to specify the percentage of each of five food groups that you want to eat (Lines 1050–1200). Note that the screen is cleared at the beginning of this routine. This allows all five food groups to be listed on the screen and the cursor to be moved beside each group so you can enter its percentage in your diet (Line 1150).

It takes different weights of various types of food to produce one calorie of energy. For example, one gram of protein produces four calories, while a gram of fat produces nine calories. Thus, depending upon how many calories of energy you want to replenish each day, you must eat a certain weight of food. This calculation of food weight (FWT) is made in Line 1340.

Because fat produces the most calories per gram, the lightest food load would consist of 100% fat. Obviously, no body could actually digest this, and it would be a disastrous choice. However, currently the program does not check for a balanced diet. Although you occasionally get indigestion (subroutine at Line 2590) it is a random occurrence and happens whether or not you have specified a balanced diet.

The main program is extremely short, 13 lines, and simply calls a series of subroutines to deal with hiking pace and mishaps (Lines 1560–1690). Currently the program checks on the hiker every three days (Line 1570). This could just as easily be done more or less often, or could be a function of distance. You might think that checking on a hiker every x miles might make more sense than every x days. However, after studying the accounts of scores of through hikers, it became apparent that far fewer things happened to faster hikers, while slower ones were plagued with problems. Thus, it seems that something happens every x days whether you are covering 8 or 28 miles a day.

Note that many mishaps are a function of date or distance or both. For example, you can't have snow in New England (Line 1660) until you reach New England (distance > 1545 miles) and until September 13 (time in days > 166).

Sometimes in a program you want events to occur periodically as a function of a variable that is not increasing uniformly. For example, in this program, you have the opportunity to establish a new hiking pace and a new diet every 400 miles. But, depending upon initial hiking pace and mishaps, you may be walking a different distance every day. Thus, we put in a counter, STV, which is multiplied by 400 and compared with your cumulative mileage every three days.

```
1680 IF D > (STV + 1) * 400 THEN GOSUB 1820
```

The variable STV is incremented by 1 in the first line of the subroutine; thus, the second time around, Line 1680 will look for 800 miles, and so on. Naturally, with this approach you will rarely hit a multiple of 400 miles on the button, but you will never go more than three days beyond the specified distance.

Many of the subroutines which deal with mishaps (Lines 2060–3320) have a counter that indicates how many times that mishap has occurred. For example, the statement in Line 2460, IF HZ(I) > 1 THEN RETURN. This is used to determine the maximum number of times that particular mishaps can occur. On the actual Trail, there is no

limit to the number of times a snake can curl up in your boot or how often you can slip on rocks. Hence, if you wish, you can eliminate some or all of the tests for values of HZ.

The remaining subroutines are more or less straightforward, although one particularly messy formula ought to be explained. It is found in Line 1250 with a variation in Line 3970, specifically:

```
CD = INT(WB * 11.5 + WB * DINPT * .3 + (WB + WP) * DINPT * .21 + WB * (15 - TW) * .22)
```

This formula calculates your daily usage of calories. The first element of calorie usage is daily body maintenance (basal metabolism) of 1150 calories per day per 100 pounds of body weight, i.e, WB (body weight in pounds) times 11.5. Actually, women have a basal metabolism of 1100 calories per 100 pounds, and men 1200; you can easily make this correction, since the program asks for your sex in Line 210.

The second element is calories burned due to walking. This is proportional to the rate of walking times the time of walking (which, of course, is distance). According to studies done at Carnegie-Mellon University, the average person burns 90 calories per 100 pounds of body weight per hour at a walking speed of 3 mph. Thus, the second element in the formula equals 0.3 times body weight (in pounds) times rate (mph) times time (hours).

The third element is the calorie usage due to climbing. Assuming a body efficiency of about 30%, you use about 110 calories in raising every 100 pounds of total weight (body weight plus pack weight) each 1000 feet in elevation. Since, on average, the Trail involves a climb of 200 feet per mile, factoring these values together leads to the third element of the equation: 0.21 times (body weight + pack weight) times distance.

The fourth element of daily calorie usage is that from non-walking chores (cooking, dishwashing, eating, dressing, writing, etc.). If most of these activities are reasonably non-strenuous, calorie usage will be about 22 calories per 100 pounds of body weight per hour. Assuming nine hours of sleep (24 - 9 = 15), we subtract walking hours from 15 (your 15 waking hours) and multiply those hours by 0.22 times body weight.

There are many interesting modifications that can be made to this program to make it more realistic, but I leave it to you to discover these and incorporate them into the program.

PROGRAM VARIABLES

A Answer to input query, numeric value
A\$ Answer to input query, string
CADD Calories added due to carrying weight of food
CAL Calories to eat per day, user input
CALOFF Calories of weight lost per day
CD Calorie usage per day
COST(n) Cost of item in pack (n = 1 to 7)
CST(n) Cost of item in pack, temporary (n = 1 to 4)
CT Total of food groups
D Distance into hike, cumulative
DB Distance when sole put on boot
DDAYS Distance walked in 3 days
DINPT Distance per day, calculated from player input
DLC(n) Distance from one location to the next (n = 1 to 21)
DM Distance per day, theoretical from player input
DMAX Distance per day, maximum due to physical condition
DY Days of food to be carried
FD(n) Diet percentage of each food group (n = 1 to 5)
FOOD Type of food resupply
FWT Food weight (oz.)
HZ(n) Hazard indicator (n = 1 to 25)
I Temporary iteration variable
ITEM\$(n) Item to pack (n = 1 to 7)
ITM\$(n) Item to pack, temporary (n = 1 to 4)
IVY Sensitivity to poison ivy, player input
J Temporary iteration variable
LC\$(n) Location name (n = 1 to 21)
MD Day of month
MO\$ Month name
PC Physical condition, player input
R(n) Reduction in walking rate due to mishaps (n = 1 to 10)
RAIN Feelings about rain, player input
RM Rate of walking multiplier
RN Random-number seed value
RT Rate-of-walking multiplier, temporary

RW Rate-of-walking, player input (mph)
 SNO Snow indicator (number of times of heavy snow)
 SP\$(n) Categories of stuff to pack (n = 1 to 7)
 STK Walking stick (0 = no, 1 = yes)
 STV Stop to get new input variable (400 - mile markers)
 T Time into hike (days)
 TD Time, temporary (days)
 TM Time delay due to mishap (days)
 TS Time since last weight-loss calculation
 TSLOW(n) Time slowed down by mishap (n = 1 to 10)
 TT Temporary indicator that date has been printed
 TW Time of walking per day, player input (hours)
 U Changes of underwear
 VL(n) Volume of item in pack (cu. in.), temporary (n = 1 to 4)
 VOL(n) Volume of item in pack (cu. in.) (n = 1 to 7)
 VOL1 Volume of tent, stove, and raingear
 VOL2 Volume of sleeping bag and mattress
 WADD Weight added to pack for food
 WB Weight, body (lbs.)
 WHT(n) Weight of item in pack, temporary (oz.) (n = 1 to 4)
 WP Weight of pack (oz. in input section, lbs. in hiking section)
 WT(n) Weight of item in pack (oz.) (n = 1 to 7)
 WTLOSS Weight loss since last weight-loss calculation
 WTLOST Weight loss, total
 WTOZ Weight in oz. (for printing only)
 WTPND Weight in lbs. (for printing only)
 X\$ Temporary string variable
 X Temporary average miles per day

PROGRAM LISTINGS

[Download APP.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF
110 LOCATE 10, 1 : X$ = "Appalachian Trail" : GOSUB 5590
120 LOCATE 13, 1 : X$ = "(c) by David H. Ahl, 1986" : GOSUB 5590
130 LOCATE 23, 1 : X$ = "Press any key to continue." : GOSUB 5590
140 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND
150 WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN : CLS : GOSUB 5620
160 DIM HZ(25), DLC(25), LC$(25)
170 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND : DB = 300 : 'New boot gives 300 extra mi
180 '
190 'Data Input Section
200 CLS : PRINT "First we need some data about you." : PRINT
210 INPUT "Your sex (male or female)";A$ : GOSUB 5540
220 IF A$ < >"M" AND A$ < >"F" THEN PRINT "Answer 'M' or 'F' please." : GOTO 210
230 INPUT "Your weight in pounds";WB
240 IF WB > 79 AND WB < 401 THEN 260
250 PRINT "Surely you jest. Let's try that one again." : GOTO 230
260 PRINT "What is your physical condition (1 = excellent, 2 = good, ";
270 INPUT "3 = fair, 4 = poor)";PC
280 IF PC < 1 OR PC > 4 THEN PRINT "Answer 1, 2, 3, or 4 please." : GOTO 260
290 PRINT : PRINT "Walking pace: You may change your pace as the hike progresses."
300 PRINT "Remember, a faster pace covers the distance more quickly but"
310 PRINT "burns more calories and has a higher risk of injury."
320 PRINT "Slow and deliberate.....1.7 mph"
330 PRINT "Moderate and vigorous.....3 mph"
340 PRINT "Fast and very difficult....4 mph"
350 INPUT "At what rate in mph do you wish to walk (number & decimal okay)";RW
360 IF RW < 1 OR RW > 4.2 THEN PRINT "A rate of" RW "mph is silly." : GOTO 350
370 PRINT : PRINT "Walking hours per day: You may change this as time goes on."
380 INPUT "To start, how many hours do you wish to walk per day";TW
390 IF TW > 14 THEN PRINT "That's just too ambitious." : GOTO 380
400 IF TW * RW < 7 THEN PRINT "You won't even reach NJ by Christmas." : GOTO 380
410 PRINT : PRINT "Your sensitivity to poison ivy:"
420 PRINT " (1) Highly sensitive" : PRINT " (2) Moderately sensitive"
430 PRINT " (3) Immune" : PRINT " (4) Had series of desensitization shots"
440 INPUT "Which number describes you";IVY
450 IF IVY < 1 OR IVY > 4 THEN PRINT "What's that? Let's try again." : GOTO 440
460 PRINT : PRINT "People who have hiked the Trail have different feelings ";
470 PRINT "about rain:" : PRINT " (1) Let it pour, I love it."

```

```

480 PRINT " (2) No problem as long as the sun comes out every few days."
490 PRINT " (3) Five solid days of rain really gets me down."
500 PRINT " (4) If I foresee a long stretch of rain, I'll hole up in a"
510 PRINT " shelter or motel and wait it out."
520 INPUT "Which number most closely describes your feeling";RAIN
530 IF RAIN < 1 OR RAIN > 4 THEN PRINT "Not possible. Again please." : GOTO 520
540 '
550 'Data on what to carry
560 PRINT : PRINT "You must make some decisions about what to pack."
570 FOR I = 1 TO 7 : READ SP$(I) : PRINT : PRINT SP$(I) ":"
580 FOR J = 1 TO 4 : READ ITM$(J), WHT(J), CST(J), VL(J) : PRINT J ".. " ITM$(J)
590 WEIGHT = WHT(J) : PRINT " "; : GOSUB 5430
600 IF VL(J) > 0 THEN PRINT ", " VL(J) "cu in";
610 IF CST(J) > 0 THEN PRINT ", price: $" CST(J)
620 NEXT J
630 INPUT "Which one do you want (number)";A
640 IF A < 1 OR A > 4 THEN PRINT "Come on now; answer 1, 2, 3, or 4" : GOTO 630
650 ITEM$(I) = ITM$(A) : WT(I) = WHT(A) : COST(I) = CST(A) : VOL(I) = VL(A) : NEXT I
660 VOL1 = VOL(1) + VOL(5) + VOL(7) : VOL2 = VOL(3) + VOL(4) : PRINT
670 IF COST(2) > 135 THEN IF VOL(2) > 3000 + VOL1 THEN 720 ELSE 690
680 IF VOL(2) > 3000 + VOL1 + VOL2 THEN 720 : 'Internal pack hold stuff?
690 PRINT "Your pack is too small to hold all those things plus clothes and"
700 PRINT "food. You'll have to take a larger pack or some smaller items."
710 PRINT : PRINT "Let's try again..." : RESTORE : GOTO 570
720 INPUT "How many changes of underwear do you want to take";U
730 IF U > 6 THEN PRINT "This is not a picnic. Take fewer." : GOTO 720
740 INPUT "Do you want to take a walking stick (Y or N)";A$ : GOSUB 5540
750 IF A$ = "Y" THEN STK = 1 ELSE STK = 0
760 PRINT : PRINT "To summarize, here is what you have chosen:"
770 FOR I = 1 TO 7 : PRINT SP$(I) " : " ITEM$(I) : NEXT I
780 PRINT "Changes of underwear:" U
790 IF STK = 1 THEN PRINT "Walking stick."
800 PRINT " In addition, you must carry (or wear) a hat, short-sleeve shirt,
810 PRINT "chamois shirt, light jacket, long underwear, hiking shorts, long"
820 PRINT "pants, 3 pairs socks, eating gear, water bottle, soap, toilet tissue,
830 PRINT "toilet supplies, towel, first-aid kit, snakebite kit, flashlight,
840 PRINT "100' nylon cord, watch, compass, lighter, bandanna, sewing kit, insect
850 PRINT "repellent, Swiss Army knife, water-purifier tablets, notebook, maps,
860 PRINT "guidebook, stuff sacks, moleskin, camera, and money." : PRINT
870 FOR I = 1 TO 7 : WP = WP + WT(I) : CST = CST + COST(I) : NEXT I : 'Summarize weights
880 WP = WP + 190 + U * 4 : IF STK = 1 THEN WP = WP + 24
890 PRINT "If you bought everything new, the total cost would be $" 225 + CST
900 PRINT "The total weight of what you are wearing and carrying is";
910 WEIGHT = WP : GOSUB 5430 : PRINT : PRINT ".... not including food or water."
920 '
930 'Data on food
940 GOSUB 5390 : PRINT : PRINT "Common systems of food supply include:"
950 PRINT " (1) Caches buried along the trail. Pros: no wasted time leaving"
960 PRINT " the Trail for food, heavy items can be buried."
970 PRINT " (2) Food sent to post offices along the way. Pros: more flexible"
980 PRINT " than caches. Cons: P.O.s closed nights, Sat pm and Sun."
990 PRINT " (3) Grocery stores and restaurants. Pros: good variety, cheap."
1000 PRINT " Cons: wasted time leaving Trail, limited opening hours."
1010 INPUT "Which will be your major method of food supply";FOOD
1020 IF FOOD < 1 OR FOOD > 3 THEN PRINT "Sorry, try again." : GOTO 1010
1030 IF FOOD = 1 THEN RT = 1 ELSE RT = .95 : 'Off-trail excursions reduce walking rate
1040 RM = RT : GOSUB 5510 : 'Short pause before screen clears
1050 CLS
1060 PRINT "Obviously, you will carry your food in the most efficient form:"
1070 PRINT "dried, dehydrated, concentrated, etc. However, you must specify"
1080 PRINT "the percentage of your diet accounted for by each of the following"
1090 PRINT "food groups (remember, all five must add up to 100).":
1100 PRINT " (1) Dairy foods, cheese, yogurt"
1110 PRINT " (2) Fruits and vegetables"
1120 PRINT " (3) Meat, poultry, fish, eggs"
1130 PRINT " (4) Bread, cereal, seeds, nuts,"
1140 PRINT " (5) Margarine, lard, oils, fats"
1150 CT = 0 : FOR I = 1 TO 5 : LOCATE I + 4, 39 : INPUT FD(I) : CT = CT + FD(I) : NEXT I
1160 LOCATE 10, 39 : PRINT CT "%" : PRINT
1170 IF CT = 100 THEN PRINT "Very good." : GOTO 1220
1180 PRINT "Sorry, but your percentages add up to" CT "rather than to 100%."

```

```

1190 PRINT : PRINT "Press any key when you're ready to try again."
1200 WHILE LEN(INKEY$) = 0 : WEND : GOTO 1050
1210 '
1220 'Calculate calorie usage
1230 DM = RW * TW : IF DM > 30 THEN DM = 30 : 'Desired distance = rate * time
1240 'Calories = metabolism + walking + climbing + camp activities
1250 CD = INT(WB * 11.5 + WB * DM * .3 + (WB + WP / 16) * DM * .21 + WB * (15 - TW) * .22)
1260 PRINT : PRINT "Given your weight and that of your supplies, your walking"
1270 PRINT "speed, and your walking time per day, you can expect"
1280 PRINT "to burn at least" CD "calories per day."
1290 PRINT : INPUT "How many calories worth of food do you want to eat"; CAL
1300 IF CAL > .6 * CD THEN 1330
1310 PRINT "Your body will rebel against burning that much body fat."
1320 PRINT "Better eat a bit more..." : GOTO 1290
1330 IF CAL > 1.5 * CD THEN PRINT "No blimps allowed on the trail." : GOTO 1290
1340 FWT = INT(CAL * 3.2 / (4 * FD(1) + 3 * FD(2) + 4 * FD(3) + 4 * FD(4) + 9 * FD(5))) :
WEIGHT = FWT
1350 PRINT "That means eating an approx food weight per day of"; : GOSUB 5430
1360 PRINT : PRINT : IF FOOD = 3 THEN DY = 2 ELSE DY = 3 : 'Avg days of food carried
1370 WADD = DY * FWT + 17 : CADD = WADD * DM * .21 : WP = WP + WADD : CD = CD + CADD
1380 PRINT "Food and water add" WADD "oz. to your trail weight bringing your"
1390 PRINT "total weight (worn and carried) to"; : WEIGHT = WP : GOSUB 5430 : PRINT
1400 PRINT : X$ = "Preparations are finally complete!" : GOSUB 5590 : GOSUB 5510
1410 '
1420 'Main Hiking Section
1430 'Initial calculations, data, and messages
1440 WP = WP / 16 : GOSUB 1930 : 'Calculate true hiking pace
1450 PRINT : PRINT : GOSUB 5100 : 'Read locations
1460 PRINT " It is April 1 and you briskly step out on the approach trail"
1470 PRINT "at Amicalola Falls, Georgia. You hike the 6.9 miles to the peak"
1480 PRINT "of Springer Mountain and sign the trail log, the first of many"
1490 PRINT "that you intend to sign. Your hike will take you through 14"
1500 PRINT "states as the Trail wanders 2007 miles along the Appalachian"
1510 PRINT "Mountains to Baxter Peak on Mt. Katahdin in Maine. It is a"
1520 PRINT "challenging trail with an average climb of 200 feet each mile."
1530 PRINT "Fewer than 2000 people have walked its entire length. Good luck!"
1540 PRINT : GOSUB 5390
1550 '
1560 'Main loop starts here
1570 T = T + 3 : TD = INT(T + .5) : PRINT : GOSUB 5200 : 'Check on the hiker every 3 days
1580 DDAYS = 3 * RM * DINPT : D = D + DDAYS : IF D > 1999 THEN 4560
1590 PRINT "You have walked" INT(D) "miles. ";
1600 FOR I = 1 TO 21 : IF D > DLC(I) - 17 AND D < DLC(I) + 17 THEN 1620 : 'Near anyplace?
1610 NEXT I : PRINT : GOTO 1630
1620 PRINT "You are near" LC$(I)
1630 IF D > 1466 THEN R(10) = .85 : TSLOW(10) = 2007 : 'Slow going in mountains
1640 GOSUB 1710 : 'Any mishaps recently?
1650 IF T > 12 THEN GOSUB 4030 : 'Long stretch of rain?
1660 IF D > 1545 AND T > 166 THEN GOSUB 4260 : 'Snow in New England
1670 IF D > 1845 AND KEN = 0 THEN GOSUB 4380 : 'Kennebec River crossing
1680 IF D > (STV + 1) * 400 THEN GOSUB 1820 : 'Allow user to reset input variables
1690 GOSUB 5390 : GOTO 1570 : 'Go back to start of hiking loop
1700 '
1710 'Subroutine to determine which mishap, if any, occurs
1720 TM = 0 : RN = INT(1 + 40 * RND(1)) : IF RN > 35 THEN RN = 36 : 'Mishap occurs
1730 IF RN > 12 THEN 1760
1740 ON RN GOSUB 2070, 2110, 2150, 2220, 2260, 2300, 2320, 2340, 2390, 2410, 2430, 2460
1750 GOTO 1800
1760 IF RN > 24 THEN 1790
1770 ON RN - 12 GOSUB 2530, 2570, 2600, 2630, 2670, 2710, 2750, 2770, 2790, 2820, 2870, 2910
1780 GOTO 1800
1790 ON RN - 24 GOSUB 2960, 2990, 3010, 3040, 3070, 3100, 3130, 3170, 3200, 3220, 3260, 3340
1800 T = T + TM : GOSUB 2020 : RETURN : 'Time delay resulting from mishap
1810 '
1820 'Subroutine to let user reset input variables
1830 STV = STV + 1 : INPUT "Want to change walking pace or hours of walking"; A$
1840 GOSUB 5540 : IF A$ < ">" THEN 1910
1850 INPUT "New walking pace (mph)"; RW
1860 IF RW < 1 OR RW > 4.5 THEN PRINT "A rate of" RW "mph is silly." : GOTO 1850
1870 INPUT "New hours per day on the trail"; TW
1880 IF TW > 14 THEN PRINT "Come now; that's just too ambitious." : GOTO 1870

```

```

1890 IF STK = 1 THEN 1910 ELSE PRINT "Want to change your mind and carry a";
1900 INPUT "walking stick";A$: GOSUB 5540 : IF A$ = "Y" THEN STK = 1
1910 PRINT : GOSUB 1930 : RETURN
1920 '
1930 'Subroutine to establish true hiking pace
1940 DINPT = RW * TW : 'Desired distance = walking rate * hours per day
1950 IF D > 600 THEN PC = 1 : 'Under 600 miles physical condition limits mileage
1960 DMAX = 6 + 6 * (5 - PC) : IF DINPT > DMAX THEN DINPT = DMAX
1970 IF WB / WP > 6 THEN 1990 : 'Body weight to pack weight ratio under 6?
1980 DINPT = (.49 + .086 * WB / WP) * DINPT : 'Heavy pack cuts down speed
1990 IF STV = 0 THEN RETURN : 'No chance to change diet at the start
2000 GOSUB 3820 : RETURN : 'Chance to change diet as trip progresses
2010 '
2020 'Subroutine to alter hiking rate due to mishaps
2030 RM = RT : FOR I = 1 TO 10 : IF TSLow(I) > T THEN RM = RM * R(I)
2040 NEXT I : RETURN
2050 '
2060 'Subroutines for 35 assorted mishaps follow
2070 IF D > 360 OR HZ(6) = 1 THEN RETURN ELSE TM = .5 : HZ(6) = 1
2080 PRINT "You run into Rangers on military exercises who advise you to avoid"
2090 PRINT "the trail for the next few miles because of booby traps." : RETURN
2100 '
2110 IF D < 800 OR HZ(12) = 1 THEN RETURN ELSE TM = 1 : HZ(12) = 1
2120 PRINT "The back-support strap on your backpack has worn through. You'll"
2130 PRINT "have to find a shoemaker to sew on a piece of heavy leather." : RETURN
2140 '
2150 IF D - DB < 500 THEN RETURN ELSE IF HZ(15) = 1 THEN 2180 ELSE TM = 1 : HZ(15) = 1
2160 PRINT "The soles of your boots have worn through. You'll have to get new"
2170 PRINT "soles at a shoemaker." : RETURN
2180 IF D - DB < 800 THEN RETURN ELSE TM = 1.5 : DB = D : HZ(15) = 0
2190 PRINT "Your repaired boot soles are going again. You'll have to buy a new"
2200 PRINT "pair of boots along the way." : RETURN
2210 '
2220 IF D - DB < 700 THEN RETURN ELSE IF HZ(24) = 1 THEN RETURN ELSE TM = 1 : HZ(24) = 1
2230 PRINT "The uppers on your boots are starting to separate from the soles."
2240 PRINT "You'll have to get them repaired by shoemaker." : RETURN
2250 '
2260 IF D < 1000 OR HZ(23) = 1 THEN RETURN ELSE TM = .4 : HZ(23) = 1
2270 PRINT "The seat of your pants has worn through. You can take off the"
2280 PRINT "pockets and sew them over the holes." : RETURN
2290 '
2300 TM = .3 : PRINT "Mice got into your food last night. Yuck." : RETURN
2310 '
2320 TM = .3 : PRINT "Some Boy Scouts kept you awake 'til 1 am last night." : RETURN
2330 '
2340 IF D > 900 THEN RETURN ELSE TM = .1
2350 PRINT "Curs attack you as you are walking through a small town."
2360 IF STK = 1 THEN PRINT "You drive them off with your walking stick." : RETURN
2370 PRINT "They nip at your heels. You should really carry a stick." : RETURN
2380 '
2390 TM = .3 : PRINT "Trail is poorly marked and you get temporarily lost." : RETURN
2400 '
2410 TM = .2 : PRINT "Broken zipper on your pack. Lose time drying stuff." : RETURN
2420 '
2430 TM = .3 : PRINT "Route marked on map is out of date. You lose time trying"
2440 PRINT "to get back on the trail." : RETURN
2450 '
2460 IF HZ(1) > 1 THEN RETURN ELSE TM = 2 : HZ(1) = HZ(1) + 1 : DOC = 1
2470 PRINT "You forget to shake out your boot and a snake has curled up inside"
2480 PRINT "for the night. You're scared and he's mad." : GOSUB 5510
2490 IF RND(1) < .9 THEN PRINT "He slithers away and all is okay. Whew!" : RETURN
2500 PRINT "It's a rattler and he bites you. You'll have to get a doctor."
2510 GOTO 3360
2520 '
2530 IF D > 165 OR HZ(7) = 1 THEN RETURN ELSE HZ(7) = 1 : RN = INT(1 + 4 * RND(1)) : TM = RN
2540 PRINT "Late snow in the Smokies. The trail is unpassable for" RN "days."
2550 RETURN
2560 '
2570 IF HZ(8) > 3 THEN RETURN ELSE HZ(8) = HZ(8) + 1 : R(1) = .9 : TSLow(1) = T + 14
2580 PRINT "You have some nasty blisters that will slow your pace." : RETURN
2590 '

```

```

2600 IF HZ(9) > 2 THEN RETURN ELSE HZ(9) = HZ(9) + 1 : R(2) = .7 : TSLOW(2) = T + 3
2610 PRINT "You have bad indigestion from an unbalanced diet." : RETURN
2620 '
2630 IF D > 870 OR HZ(10) = 1 THEN RETURN ELSE HZ(10) = 1 : TM = .5
2640 PRINT "A bear got into your food and ripped your pack last night. It's"
2650 PRINT "a good thing he wasn't hungry for human burgers." : RETURN
2660 '
2670 IF HZ(11) > 1 THEN RETURN ELSE HZ(11) = HZ(11) + 1 : R(3) = .75 : TSLOW(3) = T + 6
2680 PRINT "You twisted your ankle crossing a stream. That will slow your"
2690 PRINT "pace for a few days. Be more careful!" : RETURN
2700 '
2710 IF HZ(2) = 1 THEN RETURN ELSE HZ(2) = 1 : DOC = 2
2720 PRINT "You slipped on some rocks on a ledge. It's incredibly painful!"
2730 PRINT "Better see a doctor." : GOTO 3360
2740 '
2750 TM = .3 : PRINT "A branch snaps in your eye. Lose time to treat it." : RETURN
2760 '
2770 TM = .3 : PRINT "Bad case of constipation. Better change diet." : RETURN
2780 '
2790 TM = .5 : PRINT "You fell in a stream and everything got wet. Lose time"
2800 PRINT "drying out your sleeping bag and clothes." : RETURN
2810 '
2820 IF HZ(13) = 1 THEN RETURN ELSE HZ(13) = 1 : TM = 1.5
2830 PRINT "Last night you saw an animal moving near you and swatted at it."
2840 GOSUB 5510 : PRINT "Big mistake! It was a skunk. You'll have to wash"
2850 PRINT "what you can and replace the rest." : RETURN
2860 '
2870 IF HZ(14) = 1 THEN RETURN ELSE HZ(14) = 1 : TM = 2
2880 PRINT "Lowliest of the low! Someone stole your pack while you were taking"
2890 PRINT "a shower. You'll have to replace everything." : RETURN
2900 '
2910 IF HZ(3) = 1 THEN RETURN ELSE HZ(3) = 1 : TM = 3 : DOC = 3
2920 PRINT "After five solid days of rain, everything is soaked and you just"
2930 PRINT "can't stop shivering. You feel so terrible that you'll have to"
2940 PRINT "see a doctor." : GOTO 3360
2950 '
2960 IF HZ(16) = 1 THEN RETURN ELSE HZ(16) = 1 : TM = .4
2970 PRINT "You cut your hand badly with your knife. Be careful!" : RETURN
2980 '
2990 TM = .3 : PRINT "Socks worn through. You'll have to buy new ones." : RETURN
3000 '
3010 IF HZ(17) = 1 THEN RETURN ELSE HZ(17) = 1 : TM = 1
3020 PRINT "Bad toothache. You'll have to find a dentist soon." : RETURN
3030 '
3040 IF HZ(18) = 2 OR WT(1) = 0 THEN RETURN ELSE HZ(18) = HZ(18) + 1 : TM = .5
3050 PRINT "Tent ripped. You'll have to sew on a patch." : RETURN
3060 '
3070 IF D < 1000 OR HZ(19) = 1 THEN RETURN ELSE HZ(19) = 1 : TM = .3
3080 PRINT "Your groundcloth is in shreds. Must buy a new one." : RETURN
3090 '
3100 IF D < 500 OR HZ(20) = 1 OR WT(7) = 0 THEN RETURN ELSE HZ(20) = 1 : TM = .4
3110 PRINT "Bad rip in raingear. Must get a replacement." : RETURN
3120 '
3130 IF D < 600 OR HZ(21) = 1 OR STK = 0 THEN RETURN ELSE HZ(21) = 1 : R(4) = .9 : TM = .4
3140 PRINT "Your walking stick breaks. You can get a new one in the next"
3150 PRINT "town. Maybe they make 'em better up here." : TSLOW(4) = T + 4 : RETURN
3160 '
3170 IF HZ(22) = 1 THEN RETURN ELSE HZ(22) = 1 : TM = .4
3180 PRINT "Your water bag springs a leak. Better get a new one soon!" : RETURN
3190 '
3200 TM = .2 : PRINT "You run out of toilet tissue. Yucko!" : RETURN
3210 '
3220 PRINT "You run out of water and the springs marked on the map seem"
3230 PRINT "to have vanished or dried up. Better take it easy for a bit."
3240 HZ(5) = 1 : R(5) = .9 : TSLOW(5) = T + 3 : RETURN
3250 '
3260 IF HZ(4) = 3 THEN RETURN ELSE HZ(4) = HZ(4) + 1 : DOC = 4
3270 PRINT "Oh oh, you stumble into a thicket of poison ivy. Zowie!"
3280 IF IVY = 1 OR (IVY = 2 AND RND(1) > .5) THEN 3290 ELSE 3300
3290 PRINT "You got it really bad. You'll have to see a doctor." : GOTO 3360
3300 PRINT "Like it or not, you got a mild case. It itches like crazy but the"

```

```

3310 PRINT "calamine seems to have it under control. It slows you down tho'."
3320 R(6) = .9 : TSLow(6) = T + 7 : TM = .4 : RETURN
3330 '
3340 PRINT "Walking...walking...walking...walking." : RETURN
3350 '
3360 'Subroutine to deal with serious injuries and illnesses
3370 PRINT : PRINT "You're feeling horrible, but you found a nice country doctor."
3380 ON DOC GOTO 3410, 3480, 3570, 3640, 3700
3390 '
3400 'Rattlesnake bite
3410 PRINT "He examines your swollen leg and says," : GOSUB 5510
3420 PRINT "'Good thing you got here so quickly. I'll give you a shot of anti-"
3430 PRINT "venin but you're going to be out of commission for a good 4 days-"
3440 PRINT "and even after that you'll have to take it easy for a while.'"
3450 TM = 4 : R(7) = .7 : TSLow(7) = T + 15 : RETURN
3460 '
3470 'Broken or sprained leg
3480 PRINT "He examines your leg and says," : GOSUB 5510 : IF RND(1) > .7 THEN 3520
3490 PRINT "'That's a very nasty sprain. I'll tape it up, but you'll have to"
3500 PRINT "take it easy for at least a month.'"
3510 TM = 1.5 : R(8) = .6 : TSLow(8) = T + 30 : RETURN
3520 PRINT "'Bad news, my young friend. Your leg is broken. I'm surprised you"
3530 PRINT "got here under your own power. But this is the end of your hike."
3540 PRINT "Sorry, but maybe you can try again next year.'" : GOTO 4610
3550 '
3560 'Hypothermia
3570 PRINT "He examines you and says," : GOSUB 5510 : TM = INT(5 + 6 * RND(1))
3580 PRINT "'That prolonged rain and cold has put you in a condition that we"
3590 PRINT "call hypothermia. You can stay in town here at a motel and I'll"
3600 PRINT "keep an eye on you, but you can't go back on the Trail for at"
3610 PRINT "least" TM "days. Sorry, but that's the way it is.'" : RETURN
3620 '
3630 'Poison Ivy
3640 PRINT "He only needs a glance to see that you are suffering from an"
3650 PRINT "extremely bad case of poison ivy. He puts you in a clinic for"
3660 PRINT "a few days and tells you what you already knew—that you'll just"
3670 PRINT "have to let it run its course." : TM = 5 : RETURN
3680 '
3690 'Excessive weight loss
3700 PRINT "He examines you and says," : GOSUB 5510 : IF WTLOST > .33 * WB THEN 3770
3710 PRINT "'You may want to lose some weight, but it's coming off far too"
3720 PRINT "quickly. Your body just can't cope. I'm going to keep you here"
3730 PRINT "for a week on a controlled diet—and then for the rest of the trip"
3740 PRINT "you'll have to go at a slower pace. Also, I want you to consume"
3750 PRINT "at least as many calories per day as your body is using up.'"
3760 WB = 1.18 * WB : TM = 7 : R(9) = .8 : TSLow(9) = T + 30 : RETURN
3770 PRINT "'Believe it or not, you are in an advanced stage of starvation."
3780 PRINT "You're going to have to remain here for a few weeks on a controlled"
3790 PRINT "diet to stabilize your body chemistry. And then you will go home—"
3800 PRINT "yes, HOME, and not back to the Trail this year.'" : GOTO 4610
3810 '
3820 'Subroutine to examine weight loss
3830 CALOFF = CD - 1.03 * CAL : 'Augment diet by 3% with ice cream, etc.
3840 TS = T - TL : TL = T : 'Time (in days) of this trip segment
3850 WTLOSS = CALOFF * TS / 3500 : WTLOST = WTLOST + WTLOSS : IF WTLOSS < 1 THEN RETURN
3860 PRINT "Since the start of the trip, you have lost" INT(WTLOST) "pounds."
3870 IF WTLOST > .07 * WB THEN 3900 : 'Lost more than 7% of orig body weight?
3880 PRINT "Right now you are feeling fit as a fiddle, but remember, you"
3890 PRINT "still have a long way to go." : GOTO 3970
3900 IF WTLOST > .15 * WB THEN 3930 : 'Lost more than 15% of orig body weight?
3910 PRINT "You occasionally feel a bit lightheaded and shaky. You really"
3920 PRINT "should eat a bit more." : GOTO 3970
3930 PRINT "That's far too much weight to lose in this short period of time."
3940 IF WTLOST > .24 * WB THEN HZ(5) = 1 : DOC = 5 : GOSUB 3360 : GOTO 3970
3950 PRINT "You frequently feel lightheaded, nauseated, and sluggish. You'd"
3960 PRINT "better add to your diet...and soon!"
3970 WB = WB.WTLOSS : CD = INT(WB * 11.5 + WB * DINPT * .3 + (WB + WP) * DINPT * .21 + WB *
(15 - TW) * .22)
3980 PRINT "At your current pace, you are burning" CD "calories per day."
3990 PRINT : INPUT "How many calories worth of food do you want to eat";CAL
4000 IF CD - CAL > 400 THEN PRINT "Okay, suit yourself."

```

```

4010 RETURN
4020 '
4030 'Subroutine to deal with a long period of rain
4040 IF RND(1) < .94 THEN HZ(5) = 0 : RETURN : '6% chance of heavy rain
4050 IF HZ(5) = 1 THEN HZ(5) = 0 : RETURN : 'In a dry stretch?
4060 HZ(25) = HZ(25) + 1 : TM = RAIN * 1.7 : IF HZ(25) > 4 THEN HZ(25) = 4
4070 ON HZ(25) GOTO 4080, 4130, 4150, 4180
4080 PRINT "It has been raining steadily for the past week and you are"
4090 PRINT "thoroughly soaked—your clothes, your sleeping bag, your food,"
4100 PRINT "and, yes, your spirits. What a bummer." : GOSUB 5510
4110 PRINT "If it continues for another week, you're going to have fungus"
4120 PRINT "growing on your skin." : RETURN
4130 PRINT "Good grief! More rain—torrential, blustery, miserable rain."
4140 PRINT "This is really beginning to get you down." : RETURN
4150 PRINT "Would you believe it? It is raining again. Not the pitter-patter"
4160 PRINT "of the songwriters, but steady, heavy, cold rain." : GOSUB 5510
4170 PRINT "...and more rain. Won't it ever stop?" : RETURN
4180 PRINT "Unbelievable...it is raining again...and has been for the past week."
4190 IF (RAIN = 2 OR RAIN = 3) AND D < 1900 THEN 4230 ELSE RETURN
4200 IF RAIN = 4 THEN PRINT "Okay, you resign yourself to wait it out." : RETURN
4210 PRINT "Even your cheerful attitude toward rain is taking a beating, but"
4220 PRINT "you keep slogging along, hoping for a letup." : RETURN
4230 GOSUB 5510 : PRINT : PRINT "That's it. You can't take any more. Maybe"
4240 PRINT "you'll try again next year, but that's it for now." : GOTO 4610
4250 '
4260 'Subroutine to deal with snow in New England
4270 IF T > 200 AND RND(1) > .5 THEN 4290 : '50% chance of snow after Oct 15
4280 IF RND(1) > .2 THEN RETURN : '20% chance of snow
4290 SNO = SNO + 1 : PRINT "Oh oh, New England is getting some snow..."
4300 IF SNO = 1 THEN PRINT "but you keep pushing on." : RETURN
4310 IF SNO > 2 THEN 4350 : '3 heavy snowfalls and you're out
4320 PRINT "You pushed through the last flurries but this looks"
4330 PRINT "more serious. You say to yourself, 'I've gone this"
4340 PRINT "far, I'm going to go all the way.' And on you go..." : RETURN
4350 PRINT "You made a gallant attempt to get through, but the Park"
4360 PRINT "Rangers won't let you go on. Too bad." : GOTO 4610
4370 '
4380 'Subroutine to deal with the Kennebec River
4390 KEN = 1 : PRINT : PRINT "You have arrived at the Kennebec River."
4400 INPUT "Did you make prior arrangements to get across";A$: GOSUB 5540
4410 IF A$ < >"Y" THEN 4450 ELSE RN = RND(1) : IF RN > .5 THEN 4440
4420 PRINT "Fortunately the person you called showed up to meet you with"
4430 PRINT "a canoe. You get across in jig time." : TM = .5 : GOTO 4540
4440 GOSUB 5510 : PRINT "Too bad; the guy you called didn't show up." : GOTO 4460
4450 PRINT "That wasn't very sensible. What will you do now?" : GOSUB 5510
4460 IF RND(1) > .7 THEN 4520 : '30% chance you can ford the river
4470 PRINT "The river is running very high and the logs from the sawmill are"
4480 PRINT "very dangerous. You'll have to hang around until another hiker"
4490 PRINT "(who, hopefully, has arranged for a canoe) shows up or hope that"
4500 PRINT "someone comes along." : GOSUB 5510 : TM = INT(2 + 3 * RND(1))
4510 PRINT "Finally...you're across, but it cost you" TM "days." : GOTO 4540
4520 PRINT "Fortunately the river isn't running too high and you can probably"
4530 PRINT "wade across downstream at the ford. Boy, were you lucky!" : TM = .6
4540 T = T + TM : RETURN
4550 '
4560 'Reached end of trail!
4570 PRINT "You reached the end of the trail at Baxter Peak on Mt. Katahdin!"
4580 FOR J = 1 TO 3 : GOSUB 5510 : NEXT J : CLS : FOR J = 1 TO 10
4590 BEEP : X$ = "CONGRATULATIONS!" : LOCATE 12, 1 : GOSUB 5590
4600 FOR I = 1 TO 100 : NEXT I : CLS : FOR I = 1 TO 50 : NEXT I : NEXT J : D = 2007
4610 TD = INT(T + .5) : D = INT(D) : X = (INT(.5 + 10 * D / TD)) / 10 : 'End game statistics
4620 PRINT : PRINT "It is now "; : GOSUB 5220 : PRINT "and you have been on the"
4630 PRINT "trail for" TD "days. You have covered" D "miles. Your average"
4640 PRINT "speed, considering all the delays, was ";
4650 PRINT USING "##.##";X; : PRINT " miles per day."
4660 WB = INT(WB + .5) : WL = INT(WTLOST + .5) : IF WL > 0 THEN X$ = "less" : GOTO 4680
4670 X$ = "more" : WL = -WL
4680 PRINT "You weighed" WB "pounds at the end," WL X$ " than at the start."
4690 PRINT "Nice going!"
4700 PRINT : INPUT "Would you like to try again (Y or N)"; A$ : GOSUB 5540
4710 IF A$ = "Y" THEN RUN ELSE CLS : KEY ON : END

```



```

4720 '
4730 'Names of items that can be carried
4740 DATA "Tent"
4750 DATA "Sierra Designs Divine Light, 20 sq ft, max height 34 in.", 35, 135, 214
4760 DATA "Eureka! Crescent Bike, 22 sq ft, height 43 in.", 48, 125, 353
4770 DATA "Moss Starlet, 29 sq ft; with vestibule, 37; height 39 in.", 78, 250, 334
4780 DATA "None. Use trail shelters and sleep in open.", 0, 0, 0
4790 DATA "Pack"
4800 DATA "Kelty Mountaineer external - frame with Seneca pack", 69, 139, 3975
4810 DATA "Jansport D2 external - frame pack", 99, 169, 5520
4820 DATA "Camp Trails Grey Wolf internal - frame pack, large", 82, 119, 5975
4830 DATA "Coleman Peak 1 model 680 internal - frame pack", 58, 115, 4013
4840 DATA "Sleeping bag"
4850 DATA "North Face Blue Kazoo, mummy, goose down, rated 25 deg", 34, 140, 452
4860 DATA "Slumberjack Bike Lite, mummy, Quallofil, rated 30 deg", 50, 65, 808
4870 DATA "REI 747 Wide Body, semi-rect, Quallofil, rated 30 deg", 54, 90, 1884
4880 DATA "L.L. Bean Ultra - Lite, rectangular, Quallofil, rated 35 deg", 58, 80, 804
4890 DATA "Sleeping pad/mattress"
4900 DATA "Sevylor Superlight air mattress", 32, 20, 360
4910 DATA "Therm-A-Rest self - inflating ultra-lite pad", 28, 48, 325
4920 DATA "Ensolite 1/2 in. pad", 24, 23, 300
4930 DATA "None.", 0, 0, 0
4940 DATA "Stove"
4950 DATA "MSR Whisperlite, uses white gas (available along trail)", 18, 37, 120
4960 DATA "Gaz Bleuett, fuel: butane cartridge (light and reliable)", 16, 20, 106
4970 DATA "Primus Grasshopper, fuel: propane cylinder (long life)", 22, 19, 90
4980 DATA "None (incidentally, wood fires are PROHIBITED on the trail)", 0, 0, 0
4990 DATA "Boots"
5000 DATA "Asolo Trail II S, mid-ankle leather boots", 49, 140, 0
5010 DATA "Danner Featherlight Trail, mid-ankle leather & Gore-Tex boot", 52, 95, 0
5020 DATA "Timberland Lightweight Hiker, mid-ankle fabric and Gore-Tex", 46, 50, 0
5030 DATA "Raichle Montagnas, full-height leather boots", 80, 125, 0
5040 DATA "Raingear"
5050 DATA "Early Winters Ultralight Gore-Tex rain parka" 13, 145, 25
5060 DATA "Patagonia featherweight Gore-Tex shell" 7, 58, 15
5070 DATA "Campmor nylon poncho", 16, 25, 22
5080 DATA "None.", 0, 0, 0
5090 '
5100 FOR I = 1 TO 21 : READ DLC(I), LC$(I) : NEXT I : RETURN
5110 DATA 79, "Bly Gap, GA", 165, "Doe Knob, NC", 302, "Big Bald Mt., NC"
5120 DATA 384, "Wilbur Lake, TN", 483, "Big Walker Lookout, VA", 602, "Tinker Mt., VA"
5130 DATA 698, "Salt Log Gap, VA", 800, "Fishers Gap, VA", 889, "Potomac River, WV"
5140 DATA 966, "Piney Mt., PA", 1099, "Baer Rocks, PA", 1190, "High Point, NC"
5150 DATA 1272, "Shenandoah Mt., NY", 1361, "Sages Ravine, CT", 1435
5160 DATA "Mt. Greylock, MA", 1554, "Killington, VT", 1687, "Mt. Washington, NH"
5170 DATA 1776, "Bemis Pond, ME", 1855, "Kennebec River, ME", 1922
5180 DATA "Chairback Mt., ME", 1992, "Penobscot West Branch, ME"
5190 '
5200 'Subroutine to print the date
5210 IF TT = TD THEN RETURN : 'Printed this date already?
5220 IF TD < 31 THEN 5250 ELSE IF TD < 62 THEN 5260 ELSE IF TD < 90 THEN 5270
5230 IF TD < 121 THEN 5280 ELSE IF TD < 152 THEN 5290 ELSE IF TD < 182 THEN 5300
5240 IF TD < 213 THEN 5310 ELSE IF TD < 225 THEN 5320 ELSE GOTO 5340
5250 MO$ = "April" : MD = TD : GOTO 5330
5260 MO$ = "May" : MD = TD - 30 : GOTO 5330
5270 MO$ = "June" : MD = TD - 61 : GOTO 5330
5280 MO$ = "July" : MD = TD - 89 : GOTO 5330
5290 MO$ = "August" : MD = TD - 120 : GOTO 5330
5300 MO$ = "September" : MD = TD - 151 : GOTO 5330
5310 MO$ = "October" : MD = TD - 181 : GOTO 5330
5320 MO$ = "November" : MD = TD - 212
5330 PRINT MO$; MD; : TT = TD : RETURN
5340 PRINT : PRINT "It's November 12 and all the New England states are covered"
5350 PRINT "with snow. You have no chance of finishing the trail. Better"
5360 PRINT "luck next year. You have been out on the"
5370 TD = INT(T + .5) : D = INT(D) : X = (INT(.5 + 10 * D / TD)) / 10 : GOTO 4630
5380 '
5390 'Subroutine to temporarily break execution
5400 PRINT : X$ = "Press any key to continue" : GOSUB 5590
5410 WHILE LEN(INKEY$) = 0 : WEND : PRINT : RETURN
5420 '

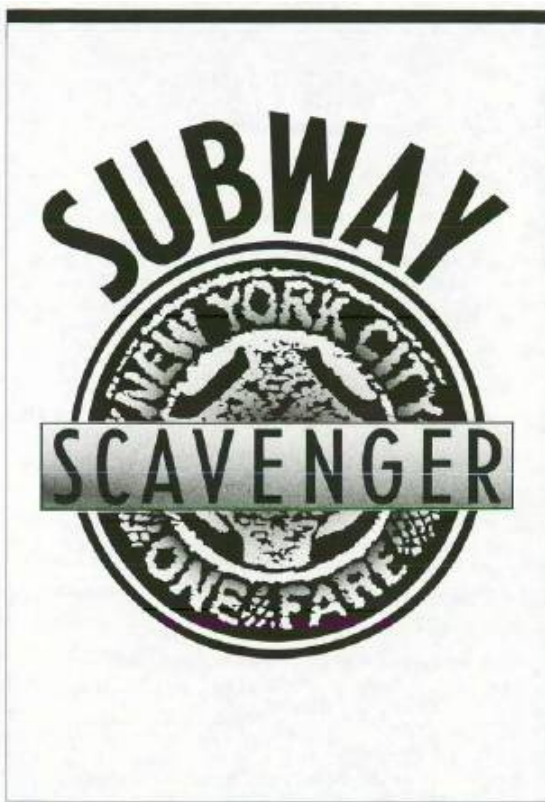
```

```

5430 'Subroutine to print weights by pounds and ounces
5440 WTPND = INT(WEIGHT / 16) : WTOZ = WEIGHT - 16 * WTPND
5450 IF WTPND > 1 THEN PRINT WTPND "pounds"; : GOTO 5470
5460 IF WTPND = 1 THEN PRINT " 1 pound"; : GOTO 5470
5470 IF WTOZ > 1 THEN PRINT WTOZ "ounces"; : RETURN
5480 IF WTOZ = 1 THEN PRINT " 1 ounce";
5490 RETURN
5500 '
5510 'Subroutine to make a short pause
5520 FOR I = 1 TO 900 : NEXT I : RETURN
5530 '
5540 'Subroutine to extract the first letter of an input answer
5550 IF A$ = "" THEN A$ = "Y" : RETURN
5560 A$ = LEFT$(A$, 1) : IF A$ >= "A" AND A$ <= "Z" THEN RETURN
5570 A$ = CHR$(ASC(A$) - 32) : RETURN
5580 '
5590 'Subroutine to print centered lines
5600 PRINT TAB((70 - LEN(X$)) / 2) X$; : RETURN
5610 '
5620 'Subroutine to print the instructions
5630 X$ = "Appalachian Trail" : GOSUB 5590 : PRINT : PRINT
5640 PRINT " You are a hiker whose goal is to walk the entire 2007 miles of"
5650 PRINT "the Appalachian Trail from Springer Mt., GA, to Mt. Katahdin, Maine."
5660 PRINT "You set out in April as soon as the Smokies are clear of snow, and"
5670 PRINT "you must reach the northern terminus before it is blocked by snow."
5680 PRINT " Your hike is divided into three-day segments. Along the way,"
5690 PRINT "you encounter natural hazards, difficulties with your equipment,"
5700 PRINT "and physical problems."
5710 PRINT " Careful planning for your hike is very important. In deciding"
5720 PRINT "what to pack, you have to make trade-offs—generally between weight"
5730 PRINT "and comfort. Of course, everything must fit in your pack."
5740 PRINT " You must decide how you will obtain food along the route, how"
5750 PRINT "much to eat in each food group, and how many calories to replenish."
5760 PRINT " You must decide at what pace you will walk, and how long to"
5770 PRINT "hike each day. Of course, a faster pace will cover mileage more"
5780 PRINT "quickly than a slower one, but it is much harder on your body."
5790 PRINT " You don't have many choices when dealing with mishaps. It is"
5800 PRINT "assumed that you are a sensible hiker, make repairs when necessary,"
5810 PRINT "replace things that wear out, and see a doctor if you get sick."
5820 PRINT "Nevertheless, mishaps cost you time, of which you have little to"
5830 PRINT "spare as you take another of the five million steps towards Maine."
5840 PRINT : X$ = "Press any key when you're ready to go." : GOSUB 5590 : RETURN

```

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)

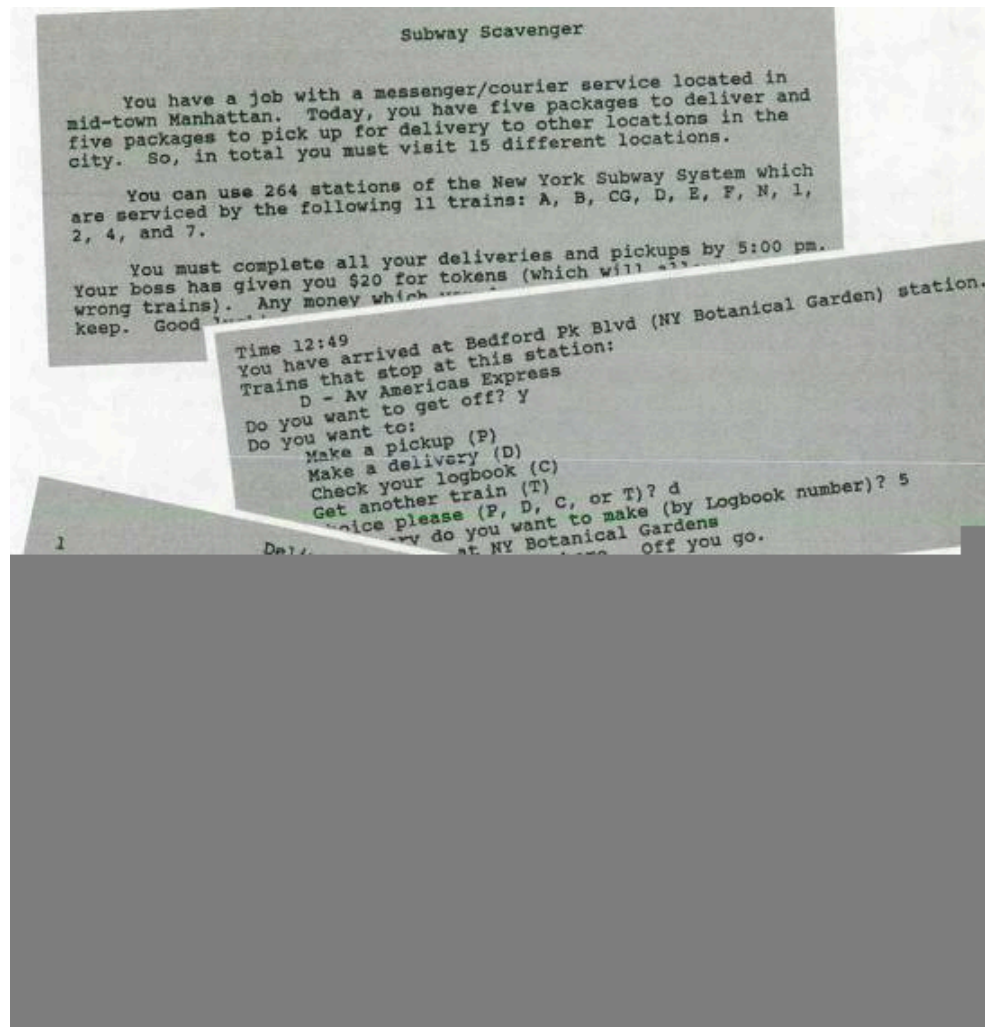


THE GAME

In Subway Scavenger, you are a messenger hired to pick up and deliver packages all over New York City. Starting at 9:00 A.M., your log will show five deliveries and five pickups. You start from your office in the Port Authority Bus Terminal at 8th Avenue and 42nd Street and work until quitting time. During the course of your day, you face typical New York subway hazards—stuck doors, track fires, and unsavory characters—but the clock is your biggest enemy. The following rules govern your progress through your delivery schedule:

- When you arrive at a station on foot, you are told which trains stop at that station. You buy a token for \$1 and board whichever train you wish.
- At each station, you can either get off or stay on. If you get off, you can then take another train (if the station is an interchange point) or make a delivery or pickup (if the station is within walking distance of your destination). Remember that express trains don't stop at all stations. The following clues should help non-New Yorkers:
 - » Brooklyn Academy of Music is abbreviated BAM.
 - » Rockaway Boulevard in Woodhaven (Queens) is on the A Train line.
 - » Harlem is serviced by the A, B, and D Lines.
 - » The World Trade Center is an easy walk from Chambers Street.
 - » The New York Stock Exchange is an easy walk from Wall Street.
 - » Grand Street and Canal Street run through Chinatown.
- The game is easier if you make a map as you go along.

By the time you complete a few games of Subway Scavenger, you will feel like a native New Yorker—and probably be qualified for a high-paying job as a subway courier. Write if you get work!



THE STORY

The first underground subway system in the world went into service in London, England in 1863. Other early subway systems include those of Glasgow (1886), Budapest (1896), Boston (the first in the United States, 1898), Paris (1900), Berlin (1902), New York (1904), Madrid (1919), Tokyo (1927), Moscow (1935) and Chicago (1943). Toronto's subway, completed in 1954, was the first in Canada.

Some subways consist of only a single line, but most, such as the Métropolitain in Paris, the New York Subway System, and the London Underground, are networks. By far the largest underground transportation system in the world is that of New York City, which includes 230 route miles of track and 465 stations. But development of the system did not occur quickly or easily.

As immigrants poured into New York in the early 1800s, the City grew so fast that street congestion soon became a major problem for pedestrians and drivers of horse-drawn wagons, omnibuses, and carriages. As a result, several mass-transit systems were proposed for New York City, with a subway suggested as early as 1864. However, it was an elevated railway that got the approval of both backers and politicians, and the first one, a cable-operated affair along Ninth Avenue, went into operation on February 14, 1870.

The cable frequently broke, disrupting service, so, on October 27, 1871, the line installed a steam locomotive to pull the cars. At the same time, a passing track was built in the center of the line, enabling trains to run in two directions. Running time over the entire route from Dey Street (near the Battery at the southern tip of Manhattan) to 29th Street was 28 minutes. The success of this line—more than 5000 fares daily—convinced a legion of skeptics that the "el" was the panacea for all of Manhattan's street-level traffic problems, and for the next 20 years the els spread like cucumber vines all over the City.

The disadvantages of the elevated railways were that they were noisy, smelly, dirty, and a detriment to the neighborhoods over which they rolled. An inventor, Frank J. Sprague, had a solution: electricity. Sprague had built an experimental electric car and won the attention of Jay Gould, the great New York railroad financier. Sprague persuaded Gould to take a ride in his experimental car. Anxious to show it off, Sprague enthusiastically yanked the controller to set the train in motion. But he pulled the controller too abruptly, and a fuse blew. The noise, which sounded like a bomb exploding, completely unnerved Gould, who instantly abandoned all interest in electric traction. Sprague was forced to look elsewhere for backing.

"Elsewhere" turned out to be Chicago, Illinois, Richmond, Indiana and St. Joseph, Missouri. Sprague's electrified, multiple-unit trains worked so well in these cities that in 1898 Brooklyn's steam-operated els began converting to electricity, and by 1903 all of New York's els were re-equipped with Sprague locomotives. More important, electric locomotives opened the possibility of building underground railways in Manhattan, an idea that had previously been dismissed because of the danger of smoke and steam in long underground runs.

Ground was broken in March 1900 for the first underground line in New York, a five-mile run from City Hall downtown up the East Side to Grand Central Terminal at 42nd Street. Rather than using the London and Glasgow method of boring an underground tunnel, New York's planners used the open-trench approach that had been employed successfully in Budapest. Using this approach, a huge, deep trench was cut; a roof was made of steel girders; and fill and paving were added above the roof. While this method was infinitely cheaper, easier, and faster than boring a tunnel, it was not popular with shopkeepers who wondered if the din and disruption would ever end. Eventually it did, and the first subway was put into service on October 27, 1904.

By this time, the promoters had vastly extended their plans, and the line, the Interborough Rapid Transit (IRT), extended from City Hall to 145th Street and Broadway in upper Manhattan. The length of the route was 13 1/2 miles, and the travel time was 26 minutes. Extensions and other lines were soon opened to meet increasing public demand for service. By 1912, a tunnel to Brooklyn was opened, and a few years later, one took trains under the Harlem River to the Bronx.

Meanwhile, the directors of the Brooklyn Rapid Transit (BRT) Company, a holding company that owned several els in Brooklyn, were enviously watching the success of the IRT. To grab a piece of the subway pie, they proposed to the city an ambitious plan to build a new subway from lower Manhattan up Broadway to the Queensboro Bridge, connecting it with tunnels to both Brooklyn and Staten Island. The directors of the IRT and newly-formed Hudson-Manhattan Tubes Company weren't happy about this, particularly when the City announced a splitting of the routes in what became known as "the Dual (IRT-BRT) Contracts." The BRT (later the BMT) benefited the most, getting 87.8 route miles all over Manhattan, Queens, and Brooklyn. The tunnel to Staten Island, incidentally, was never built.

The third big line to be built was the Independent Subway (IND), City-owned and operated right from the start. Although known for years as simply the Eighth Avenue Subway, that was a misnomer because the IND actually encompassed six basic routes: Washington Heights, Bronx-Grand Concourse, Coney Island, Queens-Manhattan, Sixth Avenue-Houston St., and Brooklyn-Queens crosstown. A seventh was added in 1939 to service the World's Fair at Flushing Meadow. Unlike the Victorian IRT with its mosaic-decorated stations or the BMT with its flamboyant rolling stock, the IND was a no-nonsense, modern line with bright, spacious stations, well-engineered cars, and speedy express runs.

In June 1940, New York City purchased the privately-owned Interborough Rapid Transit (IRT) and the Brooklyn Manhattan Transit (BMT) and combined them with the City-built Independent Subway System (IND) under the jurisdiction of the New York Transit Authority (TA). Today more than 6100 cars, connected in trains of from 3 to 12 units, operate day and night over 27 routes and make nearly 6400 trips on an average weekday. The lines use 12 bridges and 11 underwater tunnels and carry approximately one billion passengers per year.

In 1978, the New York City Transit Authority (TA) became a subsidiary of the Metropolitan Transportation Authority (MTA), the nation's largest transportation system. MTA vehicles carry nearly two billion riders per year, 6.3 million on an average weekday, or 57% of the population in its operating area. This exceeds the combined ridership of the systems of Chicago, Boston, and Philadelphia—the second, third, and fourth largest people movers in the country. With nearly 8000 rail cars, the MTA has a train fleet larger than all of America's other rapid-transit and commuter railroads combined.

The New York TA is charged with the operation of the subway system and surface bus fleet. The subway system alone employs nearly 28,000 workers and operates its own police department. Although accidents, track fires, delays, and crime regularly make the nightly news, the TA steadfastly maintains that the subway is actually one of the fastest and safest ways to travel around the city.

Just ask Rebecca Morris. Hailing from Youngstown, Ohio, with degrees in library science and English literature, Rebecca is not the sort of a person you would expect to find in the record books—particularly not for being the first woman to ride every inch of track from Brooklyn to the Bronx, from Queens to Manhattan, in one marathon odyssey. But, in the fall of 1973, she did—on 67 trains in 26 hours and 36 minutes.

Herman Rinke was the first person to tour the entire system for a single 5-cent fare. In 1940, just days before the three lines were unified under TA control, he rode for 25 hours. Since that day, more than 70 other people—recorded in an unofficial file at the TA Public Relations Department—have ridden the entire system. The 1961 subway map notes that a Flushing youth rode all the routes for a single token in 25 hours and 36 minutes. On April Fool's Day of 1966, the MIT Rapid Transit Club used a computer to route their attempt but failed to beat the best time to date, 24 hours and 56 minutes, set by Geoffrey Arnold in 1963. But on August 3, 1967, James Law and six high school buddies rode the entire system in 22 hours, 11 minutes, a time still cited in the *Guinness Book of World Records*.

People are fanatic about many facets of the subway. Bob Leon takes pictures of every transit nook and cranny. Howard and Suzanne Samuelson run an antique store devoted exclusively to transit material. Don Howard is a walking encyclopedia of transit lore, as are Hugh Dunne and Stan Fischler. Most of them look nostalgically back on earlier days—days of kerosene marker lanterns, days when people came to New York to ride the IRT, and days before graffiti.

In 1985, the TA set a goal of having 28% of the cars clean and graffiti-free. They succeeded, but just barely. Another goal for 1985 was to have working lights and loudspeakers, functioning climate control, accurate

destination signs, and readable maps on 90% of the subway fleet. While these goals—particularly with respect to maps—were not quite realized, the TA made notable strides in meeting most of them. In 1985, for example, 78 of the TA's 465 subway stations were repainted.

Of the subway fleet of 6125 cars, 760 have been in service since the 1940s. Although in the past, the TA had purchased 200 to 400 new cars per year, these purchases were severely curtailed during the City's fiscal crisis in the mid-seventies. However, a contract was recently negotiated to purchase 1375 new cars from Japan, with the older cars being phased out as the new Japanese-built R-62 cars are delivered. The first 260 cars were delivered under this contract in 1984, and they will continue to be delivered at the rate of roughly 250 per year through 1988.

When the subway first opened in 1904, the fare was a nickel, which was a relatively large amount at the turn of the century. However, the fare did not change for over 50 years, and by 1956 the 5-cent fare was a real bargain. In contrast, over the next 30 years the fare was raised eight times until it reached its current level of \$1.

As the fare has increased, so has the number of people who try to beat it by putting slugs in the turnstiles. Indeed, as of the end of 1985, the TA was collecting more than 13,000 slugs per day, resulting in a financial loss of almost \$3 million per year. In an attempt to thwart cheaters, the TA has changed the design of the token to one with a stainless-steel center that looks like a "bull's-eye," an apt name as they are targeted at slug users. In mid-1986, all 2600 turnstiles at 749 entry locations were changed to accept only the new token.

The next time you're in New York, ride the subway. There is no better or faster way to get around—and after you master Subway Scavenger you'll really know how to do it.

BIBLIOGRAPHY

Annual Report (1984). Metropolitan Transportation Authority, 347 Madison Ave., New York, NY 10017.

Fischler, Stan. *Uptown, Downtown*. New York: Hawthorne Books, 1976.

New York City Subway Map (Spring 1985). New York City Transit Authority, 370 Jay St., Brooklyn, NY 11201. (Note: If you request a subway map from the TA, include a #10 business-size self-addressed, stamped envelope.)

THE PROGRAM

At first glance, the Subway Scavenger program may look quite complicated, because of both its length and the complexity of many program statements. In fact, the program is shorter than it seems and is relatively easy to understand—as well as being exceptionally versatile and powerful.

Its versatility and power stem from its general framework, which could be used to represent almost any kind of system with intersecting nodes. In this particular version, the framework is used to represent the New York City Subway System. However, it could just as easily accept data from any other subway (London, Tokyo, Moscow, San Francisco, etc.), train (Amtrak, BritRail, etc.), bus, or, for that matter, pedestrian system (Disneyland, a large building, manufacturing plant, etc.). It could even be used to model a mixed-mode system like a university with foot, auto, and bus transportation. Nor is the framework limited to transportation problems; it could also be used to simulate electronic-message traffic over a data network or a mixed system using voice, data, and manual transmission of messages.

While the Subway Scavenger program is more than 570 lines long, more than 300 of them are data statements. Thus the program itself is of modest length. Indeed, when the player instructions, prompts, and minor subroutines are stripped out, the main program is only about 80 lines long.

Before discussing the program itself, it is important to be familiar with the three main types of data. First are the data about the nodes of the system, in this case the subway stations. To each node (or station) we must assign a number (STA), and we probably want to give it a name as well (STATION\$(STA)). We must also know how many lines of transportation (subway lines) intersect the node (STANU(STA)) and which ones they are (STATR(STA, n)).

Second, to each line of transportation, we must assign a number (TR) and possibly a name (TRAIN\$(TR)). In addition, we must know how many nodes it intersects (TRSTOP(TR)) and which ones they are (TRSTA(TR, n)). For convenience, we may also want to know the starting and ending point, so we can differentiate point-to-point routes from circular or continuous ones. In fact, the New York Subway System has only point-to-point routes, unlike the London Underground which has a mixture of circular and point-to-point routes.

Third, once the system is described, we must know what is to be transmitted (in this case, packages carried by a human being). To each package, we must assign a number (PKG) and, optionally, a name (PKGDES\$(PKG)). We

must also know where it is coming from and where it is going in terms of the nodes of the system (PKGSTA(PKG, n)). We don't really care which subway line it is carried on as long as it reaches its destination. In some cases, the building to which a package is to be delivered can be reached from more than one subway station. Naturally, it is usually preferable to use the closest station, because you spend less time walking. However, in some cases it is more efficient to walk a few extra blocks if you can reach a destination without having to ride out of your way to a station at which you can transfer to a closer line.

The main program consists of six major sections. The first one, "Arrive at station," (Lines 340–510) prints the name of the station at which you have arrived and the names of the trains that stop there. You can arrive on foot or by train. If you arrive on a train and that station is the end of the line, you must get off.

Otherwise, you can choose to get off or stay on the train.

If you get off the train, you must decide what to do next: make a pick-up, make a delivery, check your logbook, or catch another train. If you decide to catch another train, the program branches to the "Trains coming" section (Lines 530–670). Line 550 picks a random train to arrive at the station:

```
550 RN = INT(1 + STANU(STA) * RND(1))
```

Recall that STANU(STA) is the number of trains that stop at a particular station; thus, RN will be an integer between 1 and STANU(STA). In Line 560, RN is used as the argument of STATR(STA, RN) to recover the actual number of the train:

```
560 TR = STATR(STA, RN)
```

The next two lines check to see if the current station (STA) is at one end of the line or the other for train TR. If so, that station is eliminated as a possible destination for train TR. Otherwise, train TR could have as its destination either end of the line (selected in Line 590). If you decide to board a particular train, the routine in Lines 640–660 performs the important function of determining just where that train is along its route. This may sound unnecessary; after all, the program knows you are at station STA. However, station STA may be the fourth stop for train 1 and the twelfth stop for train 2, and so on; it is this index value that is determined by Lines 640–660:

```
640 FOR I = 1 TO TRSTOP(TR)
650 IF TRSTA(TR, I) < > STA THEN NEXT I
660 TRSTX = I
```

First, Line 640 iterates through all the possible stops for train TR. In the data, these are station numbers. Line 650 compares each station number (TRSTA(TR, I)) with the current station STA and, if a match is not found, goes on to the next one. When a match is found, Line 660 assigns the index value I to the train station identification index, TRSTX.

Next, the "Train travel" routine (Lines 690–730) adds or subtracts one from TRSTX depending upon which way you are traveling, and determines what the next station will be. This routine also calls the "Trip hazards" subroutine (Lines 1210–1500) which deals with sticky doors, muggers, track fires, and so on. After determining the next station, the program simply branches back to the "Arrive at station" section.

If you decide to make a pickup or delivery at a given station, the routine at Lines 750–880 determines whether or not that is a valid option (Has it been made already? Is your destination close enough to the station?) The subsequent routines simply update your logbook and check to see if all the pickups and deliveries have been made. If more deliveries remain, a list of nearby subway stations is displayed, and you must walk to one of them.

Throughout the program, two "costs" are assigned. First is the cost of tokens (\$1 each), which is added each time you arrive at a station on foot to take a train (subroutine at Lines 1100–1190). If you run out of money, the game ends.

The second "cost" is time. Activities and events that take time are travel (two to three minutes between stations, Line 720), waiting (one minute per train, Line 610), walking and delivery (two minutes per block, six minutes per delivery, Line 910), delays due to malfunctioning doors (Line 1250) and track fires (Line 1480), and lunch (Line 5210). If the total time exceeds eight hours (or nine hours in the easier version), your day has ended at 5:00 P.M. (or 6:00 P.M.), no more deliveries can be made, and the game ends.

The subroutine in Lines 5560–5650 shuffles the list of packages. Shuffling is discussed in detail in the description of the Orient Express program. However, you should note that in addition to shuffling a list of integers (PN(n)), this

program shuffles four other variables that are associated with the list (package names, destinations, etc.). Thus, we see that shuffling can be a general function that is not limited to a list of integers.

Note the music routine in Lines 5670 to 5700. This plays the first two bars of "Give My Regards to Broadway" and is explained in the chapter on the BASIC language in the back of the book.

PROGRAM VARIABLES

A Answer to input query (most often logbook no.)
A\$ Answer to input query, string
B Answer to query about station to walk to
DELTOT Total number of deliveries completed
DES Train destination (1 = north or west end of line, 2 = south or east end of line)
DORP(n) Delivery or pickup flag (n = package no.) (0 = open, 1 = delivery, 3 = pickup, 4 = done)
H\$, HP, HR Hours, for printing
I Temporary iteration variable
J Temporary iteration variable
K Temporary shuffling variable
LGMAX Highest entry in logbook to date
LGPKG(n) Logbook number (n = package no.)
LUN Lunch indicator (0 = no lunch, 1 = had lunch)
MIN Minutes of elapsed time from start
MN Minutes, for printing
PERS Person indicator (0 = on foot, 1 = on train)
PKGDES\$(n) Package destination name (n = package no.)
PKGSTA(n, s) Subway stations near package destination (n = package no., s = station no.)
PKSTDS(n, s) Distance from package destination to subway station (n = package no., s = station no.)
PKSTNU(n) Number of stations near package destination (n = package no.)
PN(n) Package number
PRLG Line printer indicator (0 = off, 1 = on)
PS Total packages
RN Random number
STA Station number
STANU(s) Number of trains that stop at station (s = station no.)
STATION\$(s) Station name (s = station no.)
STATR(s, t) Trains that stop at station (s = station no., t = train no.)
STNS Total stations
TK Token indicator (0 = company money, 1 = own money)
TKMAX Maximum amount of money available for tokens
TM Maximum minutes to play (480 hard, 540 easier)
TM\$ Maximum time to play (5:00 hard, 6:00 easier)
TOKEN Number of tokens used
TR Train number
TRDES\$(t, 2) Stations at each end of a line
TRAIN\$(t) Train name (t = train no.)
TRNS Total trains

TRSTA(t, p) Stations that train stops at (t = train no., p = stop no.)
TRSTOP(t) Number of stops made by train (t = train no.)
TRSTX Train station index
TRX Train heading (1 = south or east, -1 = north or west)

X\$ Temporary string variable

PROGRAM LISTINGS

[Download SUBWAY.BAS](#) (tokenized BASIC format)

```
100 CLS : KEY OFF
110 LOCATE 10, 1 : X$ = "Subway Scavenger" : GOSUB 5530
120 LOCATE 13, 1 : X$ = "(c) by David H. Ahl, 1986" : GOSUB 5530
130 LOCATE 23, 1 : GOSUB 5500 : GOSUB 1650
140 LOCATE 21, 1 : X$ = "(Initializing data - please be patient)" : GOSUB 5530
150 DIM PN(20), PKGDES$(20), PKSTNU(20), PKGSTA(20, 5), PKSTDS(20, 5), LGPKG(20)
160 DIM DORP(20), STATION$(300), STANU(300), STATR(300, 6)
170 DIM TRAIN$(12), TRSTOP(12), TRSTA(12, 50), TRDES$(12, 2), DUM(20)
180 PS = 20 : STNS = 264 : TRNS = 11 : 'For reading data: packages, stations, trains
```

```

190 GOSUB 5660 : LGMAX = 10 : TKMAX = 20 : 'Packages to start
200 GOSUB 1780 : GOSUB 2050 : GOSUB 4760 : 'Read data into variables
210 WHILE RN > 32767 : RN = RN - 65635! : WEND : RANDOMIZE RN
220 GOSUB 5560 : STA = 21 : 'Starting station
230 LOCATE 21, 1 : PRINT TAB(60) " " : GOSUB 5500 : PRINT
240 INPUT "Do you want to be able to deliver after 5:00 pm (easier)";A$
250 GOSUB 5300 : IF A$ = "Y" THEN TM = 540 : TM$ = "6:00" : GOTO 270
260 TM = 480 : TM$ = "5 : 00"
270 FOR I = 1 TO 5 : 'Information about deliveries
280 DORP(I) = 1 : NEXT I
290 FOR I = 6 TO 10 : 'Information about pick-ups
300 DORP(I) = 2 : NEXT I
310 CLS : PRINT "You may want to print or copy this screen for later reference."
320 PRINT : GOSUB 5390 : 'Print package log
330 '
340 'Arrive at station routine
350 GOSUB 5140 : 'Print time
360 PRINT "You have arrived at " STATION$(STA) "station."
370 PRINT "Trains that stop at this station:"
380 FOR I = 1 TO STANU(STA)
390 PRINT " " TRAIN$(STATR(STA, I)) : NEXT I
400 IF PERS = 0 THEN GOSUB 1100 : GOTO 530 : 'If on foot, buy token
410 IF STA < > TRSTA(TR, 1) AND STA < > TRSTA(TR, TRSTOP(TR)) THEN 430
420 PRINT "End of the line. You'll have to get off." : GOTO 450
430 INPUT "Do you want to get off";A$ : GOSUB 5300
440 IF A$ = "N" THEN 690 : 'If you want to stay on train, branch to train travel
450 PERS = 0 : PRINT "Do you want to:" : PRINT " Make a pickup (P)"
460 PRINT " Make a delivery (D)" : PRINT " Check your logbook (C)"
470 PRINT " Get another train (T)"
480 INPUT "Your choice please (P, D, C, or T)";A$
490 GOSUB 5350 : IF A$ = "P" OR A$ = "D" THEN 750 ELSE IF A$ = "T" THEN 530
500 IF A$ = "C" THEN GOSUB 5390 : GOTO 450
510 INPUT "Not a valid choice. Enter P, D, C, or T please.";A$ : GOTO 490
520 '
530 'Trains coming routine
540 GOSUB 5140 : 'Print time
550 RN = INT(1 + STANU(STA) * RND(1)) : 'Which train is coming?
560 TR = STATR(STA, RN)
570 IF STA = TRSTA(TR, 1) THEN DES = 2 : GOTO 600 : 'At one end of line?
580 IF STA = TRSTA(TR, TRSTOP(TR)) THEN DES = 1 : GOTO 600 : 'or the other?
590 DES = INT(1 + 2 * RND(1)) : 'Destination?
600 PRINT "Here comes the " TRAIN$(TR) " train to" TRDESS$(TR, DES)
610 MIN = MIN + 1
620 INPUT "Do you want to get on";A$ : GOSUB 5300
630 IF A$ = "N" THEN 550 : 'If don't get on train, wait for next one
640 FOR I = 1 TO TRSTOP(TR) : 'Find out where train is
650 IF TRSTA(TR, I) < > STA THEN NEXT I : PRINT "ERROR at Line 465"
660 TRSTX = I : 'Train station identification index
670 IF DES = 1 THEN TRX = -1 ELSE TRX = 1
680 '
690 'Train travel routine
700 PERS = 1 : PRINT "You are on the " TRAIN$(TR) " train to " TRDESS$(TR, DES)
710 GOSUB 1210 : 'Possible trip hazards
720 TRSTX = TRSTX + TRX : STA = TRSTA(TR, TRSTX) : MIN = MIN + INT(2 + 1.3 * RND(1))
730 GOTO 340 : 'Go to next station routine
740 '
750 'Pickup and delivery routine
760 IF A$ = "P" THEN X$ = "pickup" ELSE X$ = "delivery"
770 PRINT "Which " X$; : INPUT " do you want to make (by Logbook number)";A
780 IF DORP(A) < > 0 THEN 820
790 INPUT "That number seems to be in error. Want to check your logbook";A$
800 GOSUB 5300 : IF A$ = "Y" THEN GOSUB 5390
810 GOTO 450
820 PRINT "That " X$ " is at " PKGDESS$(A)
830 FOR I = 1 TO PKSTNU(A)
840 IF PKGSTA(A, I) = STA THEN 870
850 NEXT I : PRINT "which is too far to walk from this station."
860 PRINT "Perhaps you should try something else." : GOTO 450
870 IF PKSTDS(A, I) > 1 THEN X$ = "s" ELSE X$ = ""
880 PRINT "which is" PKSTDS(A, I) "block" X$ " from here. Off you go."
890 '

```

```

900 'Successful pickup or delivery
910 MIN = MIN + 2 * PKSTDS(A, I) + 6 : 'Add to time (2 min per block, 6 at destination)
920 DELTOT = DELTOT + 1 : IF DORP(A) = 2 THEN 950 : 'Is this a pick-up?
930 PRINT : PRINT "You find someone to sign for the package."
940 DORP(A) = 0 : GOTO 990 : 'Mark delivery completed
950 LGMAX = LGMAX + 1 : PRINT "You pick up a package and log it in as no." LGMAX
960 PRINT "The address on it is " PKGDESS(LGMAX)
970 DORP(A) = 0 : DORP(LGMAX) = 1 : GOTO 1020
980 '
990 'Check if all pickups and deliveries made
1000 IF DELTOT = 15 THEN 1580 : 'Have all deliveries been made?
1010 IF PKSTNU(A) = 1 THEN X$ = "" ELSE X$ = "S"
1020 PRINT : "From here you can walk to the following subway station" X$ " : "
1030 IF PKSTNU(A) = 1 THEN PRINT " " STATION$(PKGSTA(A, 1)) : GOTO 340
1040 FOR I = 1 TO PKSTNU(A) : 'Iterate through possible stations
1050 PRINT " " I " - " STATION$(PKGSTA(A, I)) : NEXT I
1060 INPUT "Which station do you want to go to (by number)";B
1070 IF B < 1 OR B > PKSTNU(A) THEN PRINT "Not a valid response." : GOTO 1060
1080 STA = PKGSTA(A, B) : MIN = MIN + 3 + PKSTDS(A, B) : GOTO 340
1090 '
1100 'Buy token subroutine
1110 TOKEN = TOKEN + 1 : IF TOKEN <= TKMAX THEN 1190
1120 PRINT : PRINT "You have spent the entire $20 your boss gave you on tokens."
1130 IF TK = 0 THEN 1150 : 'Used own money yet?
1140 PRINT "Moreover, you have used up your own money as well." : GOTO 1520
1150 TK = 1 : INPUT "Do you want to buy tokens with your own money";A$ : GOSUB 5300
1160 IF A$="N" THEN PRINT "Okay, that's it then." : GOTO 1520
1170 RN = INT(300 + 600 * RND(1)) / 100 : PRINT "You have exactly $" RN;
1180 PRINT "so you can buy" INT(RN) "more tokens." : TKMAX = TKMAX + INT(RN)
1190 RETURN
1200 '
1210 'Trip hazards subroutine
1220 'Door refuses to close
1230 IF RND(1) > .05 THEN 1290 : '5% chance of a sticky door
1240 PRINT "One of the car doors refuses to close and the train can't move."
1250 RN = INT(1 + 2.5 * RND(1)) : MIN = MIN + RN : IF RN > 1 THEN X$ = "s" ELSE X$ = ""
1260 PRINT "You're stuck here for" RN "minute" X$ "."
1270 '
1280 'Possible mugging
1290 IF RND(1) > .35 THEN 1430 : '35 - 65% chance of mugging or fire on the tracks
1300 IF RND(1) > .05 THEN RETURN : '5% chance of tough characters
1310 PRINT "Some real unsavory types are whooping it up in the car across from"
1320 INPUT "your seat. Do you want to move to another car";A$ : GOSUB 5300
1330 IF A$ = "Y" THEN IF RND(1) > .05 THEN 1350 ELSE GOTO 1370
1340 IF RND(1) > .05 THEN 1360 ELSE GOTO 1380
1350 PRINT "They jeer at you but let you pass. All is okay...for now." : RETURN
1360 PRINT "They look at you and try to bait you, but you avoid them." : RETURN
1370 PRINT "Uh oh. Two of them get up and block your way."
1380 PRINT "Oh my, oh my. They're all moving to surround you."
1390 PRINT "They pull knives and demand your money." : GOSUB 5470
1400 PRINT "You, deciding that discretion is the better part of valor, give"
1410 PRINT "them all your money and call it quits for the day." : GOTO 1520
1420 '
1430 'Fire on the track
1440 IF RND(1) > 8.000001E - 03 THEN RETURN : '0.8% chance of fire on the tracks
1450 PRINT "Uh oh. The train is slowing down and seems to be stopping."
1460 GOSUB 5470 : PRINT "You're stuck here in the tunnel." : GOSUB 5470
1470 PRINT "A trainman finally comes through and announces, 'It's just a"
1480 RN = INT(10 + 35 * RND(1)) : MIN = MIN + RN
1490 PRINT "fire on the tracks folks. We'll be under way in a few minutes.'"
1500 PRINT "In fact, the delay is more like" RN "minutes!" : RETURN
1510 '
1520 'End of game routine
1530 IF DELTOT = 15 THEN 1580 : 'Were all deliveries made?
1540 PRINT : PRINT "You made it to" DELTOT "Locations, but"
1550 PRINT "your log still shows the following items:" : GOSUB 5390
1560 GOSUB 5250 : PRINT "Perhaps you'll be able to do better tomorrow."
1570 GOTO 1610
1580 GOSUB 5250 : PRINT : PRINT TAB(25) "CONGRATULATIONS !" : PRINT
1590 PRINT "You made all your deliveries and pick-ups successfully in the"
1600 PRINT "Largest city in the world. Very good!"

```

```

1610 PRINT "You used $" TOKEN " for tokens."
1620 PRINT : INPUT "Would you like to try again";A$ : GOSUB 5300
1630 IF A$ = "Y" THEN RUN ELSE CLS : KEY ON : PRINT "Bye for now." : END
1640 '
1650 CLS : X$ = "Subway Scavenger" : GOSUB 5530 : PRINT : PRINT
1660 PRINT " You have a job with a messenger/courier service located in"
1670 PRINT "mid-town Manhattan. Today, you have five packages to deliver and"
1680 PRINT "five packages to pick up for delivery to other locations in the"
1690 PRINT "city. So, in total you must visit 15 different locations." : PRINT
1700 PRINT " You can use 264 stations of the New York Subway System which"
1710 PRINT "are serviced by the following 11 trains: A, B, CG, D, E, F, N, 1,"
1720 PRINT "2, 4, and 7." : PRINT
1730 PRINT " You must complete all your deliveries and pickups by 5:00 pm."
1740 PRINT "Your boss has given you $20 for tokens (which will allow for a few"
1750 PRINT "wrong trains). Any money that you don't use on tokens is yours to"
1760 PRINT "keep. Good luck! (You'll need it.)" : RETURN
1770 '
1780 'Subroutine to read data about package deliveries
1790 FOR I = 1 TO PS
1800 READ PN(I), PKGDES$(I), PKSTNU(I)
1810 FOR J = 1 TO PKSTNU(I)
1820 READ PKGSTA(I, J), PKSTDS(I, J)
1830 NEXT J : NEXT I : RETURN
1840 DATA 1, "Curator, Museum of Natural History", 1, 17, 1
1850 DATA 2, "George Washington Bridge Bus Terminal", 1, 5, 1
1860 DATA 3, "West Side Tennis Club, Forest Hills", 1, 75, 4
1870 DATA 4, "Nathan's at Coney Island Amusement Park", 1, 95, 2
1880 DATA 5, "Big Al's Discount Mart, Rockaway Blvd, Woodhaven", 3, 50, 1, 49, 9, 51, 9
1890 DATA 6, "Apollo Theater, 125th St, Harlem", 2, 11, 1, 12, 9
1900 DATA 7, "Met's Dugout, Shea Stadium", 1, 260, 3
1910 DATA 8, "Press Room, Yankee Stadium", 1, 246, 3
1920 DATA 9, "Lion Keeper, Bronx Zoo", 2, 204, 5, 205, 8
1930 DATA 10, "Borough Hall, Brooklyn", 2, 32, 1, 221, 2
1940 DATA 11, "Brooklyn Academy of Music", 1, 67, 2
1950 DATA 12, "Registrar, Brooklyn College, Flatbush", 1, 234, 1
1960 DATA 13, "Computer Science Dept, NYU, Washington Sq", 1, 25, 3
1970 DATA 14, "NY Botanical Gardens", 1, 55, 4
1980 DATA 15, "Windows on the World, World Trade Center", 3, 29, 1, 28, 5, 191, 1
1990 DATA 16, "Metropolitan Museum of Art", 1, 249, 1
2000 DATA 17, "Computer Education Dept, Columbia Univ.", 2, 174, 2, 175, 8
2010 DATA 18, "Alice Tully Hall, Lincoln Center", 1, 181, 2
2020 DATA 19, "New York Stock Exchange", 2, 219, 2, 252, 2
2030 DATA 20, "Lin Chows, Mott St, Chinatown", 2, 65, 4, 146, 4
2040 '
2050 'Subroutine to read data about subway stations
2060 FOR I = 1 TO STNS : 'STNS = number of subway stations
2070 READ STA, STATION$(I), STANU(I)
2080 FOR J = 1 TO STANU(I)
2090 READ STATR(I, J) : 'Read train numbers that stop at station
2100 NEXT J : NEXT I : RETURN
2110 DATA 1, "207 St/Bdwy/Wash Hts (Manhattan)", 1, 1
2120 DATA 2, "Dyckman St/Bdwy", 1, 1
2130 DATA 3, "190 St/Ft Wash Av", 1, 1
2140 DATA 4, "181 St/Ft Wash Av", 1, 1
2150 DATA 5, "175 St/GW Bridge", 1, 1
2160 DATA 6, "168 St/Bdwy (Manhattan)", 3, 1, 3, 6
2170 DATA 7, "163 St/Amsterdam Av", 1, 3
2180 DATA 8, "155 St/St Nicholas Av", 1, 3
2190 DATA 9, "145 St/St Nicholas Av", 3, 1, 3, 4
2200 DATA 10, "135 St/St Nicholas Av", 1, 3
2210 DATA 11, "125 St/St Nicholas Av", 3, 1, 3, 4
2220 DATA 12, "116 St/8 Av", 1, 3
2230 DATA 13, "110 St/Cathedral Pkwy", 1, 3
2240 DATA 14, "103 St/Central Pk W", 1, 3
2250 DATA 15, "96 St/Central Pk W", 1, 3
2260 DATA 16, "86 St/Central Pk W", 1, 3
2270 DATA 17, "81 St/Museum Natl History", 1, 3
2280 DATA 18, "72 St/Central Pk W", 1, 3
2290 DATA 19, "59 St/Columbus Circle", 4, 1, 3, 4, 6
2300 DATA 20, "50 St/8 Av", 1, 2
2310 DATA 21, "42 St/8 Av", 2, 1, 2

```

2320 DATA 22, "34 St/Penn Station", 2, 1, 2
2330 DATA 23, "23 St/8 Av", 1, 2
2340 DATA 24, "14 St/8 Av", 2, 1, 2
2350 DATA 25, "W 4 St/Washington Sq", 5, 1, 2, 3, 4, 5
2360 DATA 26, "Spring St/6 Av", 1, 2
2370 DATA 27, "Canal St/6 Av", 2, 1, 2
2380 DATA 28, "Chambers St/Church St", 3, 1, 2, 7
2390 DATA 29, "World Trade Center", 1, 2
2400 DATA 30, "Bdwy/Nassau St/Fulton St (Manhattan)", 3, 1, 7, 8
2410 DATA 31, "High St/Brooklyn Br (Bklyn)", 1, 1
2420 DATA 32, "Jay St/Borough Hall", 2, 1, 5
2430 DATA 33, "Hoyt St", 2, 1, 11
2440 DATA 34, "Lafayette Av", 1, 11
2450 DATA 35, "Clinton Av", 1, 11
2460 DATA 36, "Franklin Av", 1, 11
2470 DATA 37, "Nostrand Av", 1, 1
2480 DATA 38, "Kingston Av", 1, 11
2490 DATA 39, "Utica Av", 1, 1
2500 DATA 40, "Ralph Av", 1, 11
2510 DATA 41, "Rockaway Av", 1, 11
2520 DATA 42, "Bdwy, E NY", 1, 11
2530 DATA 43, "Liberty Av", 1, 11
2540 DATA 44, "Van Sicien Av", 1, 11
2550 DATA 45, "Shepherd Av", 1, 11
2560 DATA 46, "Euclid Av", 1, 1
2570 DATA 47, "Grant Av (Brooklyn)", 1, 1
2580 DATA 48, "80 St/Liberty Av (Queens)", 1, 1
2590 DATA 49, "88 St/Liberty Av", 1, 1
2600 DATA 50, "Rockaway Blvd", 1, 1
2610 DATA 51, "104 St/Liberty Av", 1, 1
2620 DATA 52, "111 St/Liberty Av", 1, 1
2630 DATA 53, "Lefferts Blvd (Queens)", 1, 1
2640 DATA 54, "205 St/Bainbridge Av (Bronx)", 1, 4
2650 DATA 55, "Bedford Pk Blvd (NY Botanical Garden)", 1, 4
2660 DATA 56, "Kingsbridge Rd", 1, 4
2670 DATA 57, "Fordham Rd", 1, 4
2680 DATA 58, "Tremont Av", 1, 4
2690 DATA 59, "47-50 St/Rockefeller Center", 3, 3, 4, 5
2700 DATA 60, "42 St/Av Americas", 4, 3, 4, 5, 9
2710 DATA 61, "34 St/Herald Sq", 4, 3, 4, 5, 10
2720 DATA 62, "23 St/Av Americas", 1, 5
2730 DATA 63, "14 St/Av Americas", 3, 5, 6, 7
2740 DATA 64, "Bdwy/Lafayette St", 4, 3, 4, 5, 8
2750 DATA 65, "Grand St (Manhattan)", 2, 3, 4
2760 DATA 66, "DeKalb Av/Flatbush Av (Bklyn)", 2, 4, 10
2770 DATA 67, "Atlantic Av/Pacific St/BAM", 5, 3, 4, 7, 8, 10
2780 DATA 68, "179 St/Hillside Av (Queens)", 2, 2, 5
2790 DATA 69, "169 St", 1, 2
2800 DATA 70, "Parsons Blvd", 2, 2, 5
2810 DATA 71, "Sutphin Av", 1, 2
2820 DATA 72, "Van Wyck Blvd", 1, 2
2830 DATA 73, "Union Tpk", 2, 2, 5
2840 DATA 74, "75 Av", 1, 2
2850 DATA 75, "71 Av/Continental Av/Forest Hills", 4, 2, 5, 10, 11
2860 DATA 76, "Roosevelt Av", 5, 2, 5, 9, 10, 11
2870 DATA 77, "Queens Plaza", 4, 2, 5, 10, 11
2880 DATA 78, "23 St/Ely Av (Queens)", 2, 2, 5
2890 DATA 79, "Lexington Av (Manhattan)", 2, 2, 5
2900 DATA 80, "5th Av/53 St", 2, 2, 5
2910 DATA 81, "7th Av/53 St", 3, 2, 3, 4
2920 DATA 82, "36 St/4 Av", 2, 3, 10
2930 DATA 83, "9 Av/39 St", 1, 3
2940 DATA 84, "Ft Hamilton Pky", 1, 3
2950 DATA 85, "50 St/New Utrecht Av", 1, 3
2960 DATA 86, "55 St/New Utrecht Av", 1, 3
2970 DATA 87, "62 St/New Utrecht Av", 2, 3, 10
2980 DATA 88, "71 St/New Utrecht Av", 1, 3
2990 DATA 89, "79 St/New Utrecht Av", 1, 3
3000 DATA 90, "18 Av/New Utrecht Av", 1, 3
3010 DATA 91, "20 Av/86 St", 1, 3
3020 DATA 92, "Bay Pky/86 St", 1, 3

3030 DATA 93, "25 Av/86 St", 1, 3
3040 DATA 94, "Bay 50 St", 1, 3
3050 DATA 95, "Coney Island/Surf Av (Bklyn)", 4, 3, 4, 5, 10
3060 DATA 96, "67 Av/Queens Blvd", 2, 10, 11
3070 DATA 97, "63 Dr/Queens Blvd", 2, 10, 11
3080 DATA 98, "Woodhaven Blvd", 2, 10, 11
3090 DATA 99, "Grand Av/Queens Blvd", 2, 10, 11
3100 DATA 100, "Elmhurst Av", 2, 10, 11
3110 DATA 101, "65 St/Bdwy", 2, 10, 11
3120 DATA 102, "Northern Blvd", 2, 10, 11
3130 DATA 103, "46 St/Bdwy", 2, 10, 11
3140 DATA 104, "Steinway St", 2, 10, 11
3150 DATA 105, "2 Av/Houston St", 1, 5
3160 DATA 106, "Delancey St", 1, 5
3170 DATA 107, "East Bdwy (Manhattan)", 1, 5
3180 DATA 108, "York St/Jay St (Brooklyn)", 1, 5
3190 DATA 109, "Bergen St", 2, 5, 11
3200 DATA 110, "Carroll St", 2, 5, 11
3210 DATA 111, "Smith St", 2, 5, 11
3220 DATA 112, "4 Av/9 St", 2, 5, 10
3230 DATA 113, "7 Av/9 St", 1, 5
3240 DATA 114, "15 St/Prospect Park", 1, 5
3250 DATA 115, "Ft Hamilton Pwy", 1, 5
3260 DATA 116, "Church Av", 1, 5
3270 DATA 117, "Ditmas Av", 1, 5
3280 DATA 118, "18 Av/McDonald Av", 1, 5
3290 DATA 119, "Kings Hwy", 1, 5
3300 DATA 120, "Avenue U", 1, 5
3310 DATA 121, "Avenue X", 1, 5
3320 DATA 122, "Neptune Av", 1, 5
3330 DATA 123, "W 8th/NY Aquarium", 1, 5
3340 DATA 124, "7 Av/Flatbush Av", 1, 4
3350 DATA 125, "Prospect Park", 1, 4
3360 DATA 126, "Church Av/E 18 St", 1, 4
3370 DATA 127, "Newkirk Av", 1, 4
3380 DATA 128, "Kings Hwy/E 16 St", 1, 4
3390 DATA 129, "Sheepshead Bay", 1, 4
3400 DATA 130, "Brighton Beach", 1, 4
3410 DATA 131, "Court Square", 1, 11
3420 DATA 132, "21 St/Jackson Av (Queens)", 1, 11
3430 DATA 133, "Greenpoint Av (Bklyn)", 1, 11
3440 DATA 134, "Nassau Av", 1, 11
3450 DATA 135, "Metropolitan Av", 1, 11
3460 DATA 136, "Broadway/Union Av", 1, 11
3470 DATA 137, "Flushing-Marcy Avs", 1, 11
3480 DATA 138, "Myrtle-Willoughby Avs", 1, 11
3490 DATA 139, "Bedford-Nostrand Avs", 1, 11
3500 DATA 140, "36 St/Northern Blvd", 2, 10, 11
3510 DATA 141, "Lexington Av/59-60 Sts (Manhattan)", 2, 8, 10
3520 DATA 142, "5th Av/59-60 Sts", 1, 10
3530 DATA 143, "57 St/7 Av", 1, 10
3540 DATA 144, "Times Sq/42 St/Bdwy", 4, 6, 7, 9, 10
3550 DATA 145, "Union Sq/14 St", 2, 8, 10
3560 DATA 146, "Canal St/Bdwy (Manhattan)", 1, 10
3570 DATA 147, "Union St/4 Av", 1, 10
3580 DATA 148, "Prospect Av", 1, 10
3590 DATA 149, "25 St/4 Av", 1, 10
3600 DATA 150, "45 St/4 Av", 1, 10
3610 DATA 151, "53 St/4 Av", 1, 10
3620 DATA 152, "59 St/4 Av", 1, 10
3630 DATA 153, "8 Av/62 St", 1, 10
3640 DATA 154, "Ft Hamilton Pwy", 1, 10
3650 DATA 155, "18 Av/64 St", 1, 10
3660 DATA 156, "20 Av/64 St", 1, 10
3670 DATA 157, "Bay Pwy/Av O", 1, 10
3680 DATA 158, "Kings Hwy/W 7 St", 1, 10
3690 DATA 159, "Avenue U/W 7 St", 1, 10
3700 DATA 160, "86 St/W 7 St", 1, 10
3710 DATA 161, "242 St/Van Cortlandt Park", 1, 6
3720 DATA 162, "238 St/Bdwy", 1, 6
3730 DATA 163, "231 St/Bdwy (Bronx)", 1, 6

3740 DATA 164, "225 St/Bdwy (Manhattan)", 1, 6
3750 DATA 165, "215 St/10 Av", 1, 6
3760 DATA 166, "207 St/10 Av", 1, 6
3770 DATA 167, "Dyckman Av", 1, 6
3780 DATA 168, "191 St/St Nicholas Av", 1, 6
3790 DATA 169, "181 St/St Nicholas Av", 1, 6
3800 DATA 170, "157 St/Bdwy", 1, 6
3810 DATA 171, "145 St/Bdwy", 1, 6
3820 DATA 172, "137 St/Bdwy", 1, 6
3830 DATA 173, "125 St/Bdwy", 1, 6
3840 DATA 174, "116 St/Bdwy/Columbia Univ", 1, 6
3850 DATA 175, "110 St/Cathedral Pkwy", 1, 6
3860 DATA 176, "103 St/Bdwy", 1, 6
3870 DATA 177, "96 St/Bdwy", 2, 6, 7
3880 DATA 178, "86 St/Bdwy", 1, 6
3890 DATA 179, "79 St/Bdwy", 1, 6
3900 DATA 180, "72 St/Bdwy", 2, 6, 7
3910 DATA 181, "66 St/Bdwy/Lincoln Center", 1, 6
3920 DATA 182, "50 St/Bdwy", 1, 6
3930 DATA 183, "Penn Station/34 St/7 Av", 2, 6, 7
3940 DATA 184, "28 St/7 Av", 1, 6
3950 DATA 185, "23 St/7 Av", 1, 6
3960 DATA 186, "18 St/7 Av", 1, 6
3970 DATA 187, "Christopher St", 1, 6
3980 DATA 188, "Houston St", 1, 6
3990 DATA 189, "Canal & Varick Sts", 1, 6
4000 DATA 190, "Franklin St", 1, 6
4010 DATA 191, "Chambers St/W Bdwy", 2, 6, 7
4020 DATA 192, "Cortlandt St/World Trade Center", 1, 6
4030 DATA 193, "Rector St/Greenwich St", 1, 6
4040 DATA 194, "South Ferry/Battery Park", 1, 6
4050 DATA 195, "241 St/White Plains Rd (Bronx)", 1, 7
4060 DATA 196, "238 St/White Plains Rd", 1, 7
4070 DATA 197, "233 St/White Plains Rd", 1, 7
4080 DATA 198, "225 St/White Plains Rd", 1, 7
4090 DATA 199, "219 St/White Plains Rd", 1, 7
4100 DATA 200, "Gun Hill Rd/White Plains Rd", 1, 7
4110 DATA 201, "Burke Av/White Plains Rd", 1, 7
4120 DATA 202, "Allerton Av/White Plains Rd", 1, 7
4130 DATA 203, "Pelham Pkwy/White Plains Rd", 1, 7
4140 DATA 204, "Bronx Pk E/White Plains Rd", 1, 7
4150 DATA 205, "E 180 St/Bronx Zoo", 1, 7
4160 DATA 206, "E Tremont Av/Boston Rd", 1, 7
4170 DATA 207, "174 St/Southern Blvd", 1, 7
4180 DATA 208, "Freeman St", 1, 7
4190 DATA 209, "Simpson St", 1, 7
4200 DATA 210, "Intervale Av", 1, 7
4210 DATA 211, "Prospect Av", 1, 7
4220 DATA 212, "Jackson Av", 1, 7
4230 DATA 213, "3 Av/149 St", 1, 7
4240 DATA 214, "149 St/Grand Concourse (Bronx)", 2, 7, 8
4250 DATA 215, "135 St/Lenox Av (Manhattan)", 1, 7
4260 DATA 216, "125 St/Lenox Av", 1, 7
4270 DATA 217, "116 St/Lenox Av", 1, 7
4280 DATA 218, "110 St/Lenox Av", 1, 7
4290 DATA 219, "Wall St (Manhattan)", 1, 7
4300 DATA 220, "Clark St (Brooklyn)", 1, 7
4310 DATA 221, "Borough Hall/Court St (Bklyn)", 2, 7, 8
4320 DATA 222, "Hoyt St/Fulton St", 1, 7
4330 DATA 223, "Nevins St", 2, 7, 8
4340 DATA 224, "Bergen St", 1, 7
4350 DATA 225, "Grand Army Plaza, Prospect Park", 1, 7
4360 DATA 226, "Eastern Pkwy/Brooklyn Museum", 1, 7
4370 DATA 227, "Franklin Av/Eastern Pkwy", 1, 7
4380 DATA 228, "President St", 1, 7
4390 DATA 229, "Sterling St/Nostrand Av", 1, 7
4400 DATA 230, "Winthrop St/Nostrand Av", 1, 7
4410 DATA 231, "Church Av/Nostrand Av", 1, 7
4420 DATA 232, "Beverley Rd/Nostrand Av", 1, 7
4430 DATA 233, "Newkirk Av/Nostrand Av", 1, 7
4440 DATA 234, "Flatbush Av/Bklyn College", 1, 7

```

4450 DATA 235, "Woodlawn/Jerome Av (Bronx)", 1, 8
4460 DATA 236, "Mosholu Pkwy", 1, 8
4470 DATA 237, "Bedford Park Blvd", 1, 8
4480 DATA 238, "Kingsbridge Rd", 1, 8
4490 DATA 239, "Fordham Rd/Jerome Av", 1, 8
4500 DATA 240, "183 St/Jerome Av", 1, 8
4510 DATA 241, "Burnside Av/Jerome Av", 1, 8
4520 DATA 242, "176 St/Jerome Av", 1, 8
4530 DATA 243, "Mt Eden Av/Jerome Av", 1, 8
4540 DATA 244, "170 St/Jerome Av", 1, 8
4550 DATA 245, "167 St/River Av", 1, 8
4560 DATA 246, "161 St/Yankee Stadium (Bronx)", 2, 4, 8
4570 DATA 247, "", 1, 11
4580 DATA 248, "125 St/Lexington Av (Manhattan)", 1, 8
4590 DATA 249, "86 St/Lexington Av/Metropolitan Museum", 1, 8
4600 DATA 250, "42 St/Grand Central Sta", 2, 8, 9
4610 DATA 251, "Bklyn Bridge/Worth St", 1, 8
4620 DATA 252, "Wall St/Bdwy", 1, 8
4630 DATA 253, "Bowling Green (Manhattan)", 1, 8
4640 DATA 254, "Nostrand Av/Eastern Pkwy", 1, 8
4650 DATA 255, "Rockaway Av/Livonia Av", 1, 8
4660 DATA 256, "New Lots Av (Brooklyn)", 1, 8
4670 DATA 257, "Queensboro Plaza (Queens)", 1, 9
4680 DATA 258, "61 St/Roosevelt Av", 1, 9
4690 DATA 259, "Junction Blvd", 1, 9
4700 DATA 260, "Willets Point/Shea Stadium", 1, 9
4710 DATA 261, "Main St/Flushing (Queens)", 1, 9
4720 DATA 262, "Classon Av", 1, 11
4730 DATA 263, "Clinton-Washington Avs", 1, 11
4740 DATA 264, "Fulton St/Lafayette Av", 1, 11
4750 '
4760 'Subroutine to read data about subway trains
4770 FOR I = 1 TO TRNS : 'TRNS = number of subway trains
4780 READ TR, TRAIN$(I), TRSTOP(I)
4790 FOR J = 1 TO TRSTOP(I)
4800 READ TRSTA(I, J) : 'Read station numbers for a train
4810 NEXT J
4820 TRDES$(TR, 1) = STATION$(TRSTA(TR, 1)) : 'Station name at north or west end
4830 TRDES$(TR, 2) = STATION$(TRSTA(TR, TRSTOP(TR))) : 'Station at south or east end
4840 NEXT I : RETURN
4850 DATA 1, "A - 8 Av Express", 29, 1, 2, 3, 4, 5, 6, 9, 11, 19, 21, 22, 24, 25, 27, 28
4860 DATA 30, 31, 32, 33, 37, 39, 46, 47, 48, 49, 50, 51, 52, 53
4870 DATA 2, "E - 8 Av Local", 24, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
20, 21
4880 DATA 22, 23, 24, 25, 26, 27, 28, 29
4890 DATA 3, "B - Av Americas Express", 36, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18
4900 DATA 19, 81, 59, 60, 61, 25, 64, 65, 67, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93,
94, 95
4910 DATA 4, "D - Av Americas Express", 26, 54, 55, 56, 57, 58, 246, 9, 11, 19, 81, 59, 60
4920 DATA 61, 25, 64, 65, 66, 67, 124, 125, 126, 127, 128, 129, 130, 95
4930 DATA 5, "F - Av Americas Local", 37, 68, 70, 73, 75, 76, 77, 78, 79, 80
4940 DATA 59, 60, 61, 62, 63, 25, 64, 105, 106, 107, 108, 32, 109, 110
4950 DATA 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 95
4960 DATA 6, "1 - Bdwy-7th Av Local", 38, 161, 162, 163, 164, 165, 166, 167, 168, 169, 6
4970 DATA 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 19, 182, 144, 183, 184
4980 DATA 185, 186, 63, 187, 188, 189, 190, 191, 192, 193, 194
4990 DATA 7, "2 - 7th Av Express", 49, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204
5000 DATA 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 177, 180, 144
5010 DATA 183, 63, 191, 28, 30, 219, 220, 221, 222, 223, 67, 224, 225, 226, 227, 228, 229,
230
5020 DATA 231, 232, 233, 234
5030 DATA 8, "4 - Lexington Av Express", 29, 235, 236, 237, 238, 239, 240, 241, 242, 243
5040 DATA 244, 245, 246, 214, 248, 249, 141, 250, 145, 64, 251, 30, 252, 253, 221, 223, 67,
254
5050 DATA 255, 256
5060 DATA 9, "7 - Flushing Express", 9, 144, 60, 250, 257, 258, 76, 259, 260, 261
5070 DATA 10, "N - Broadway Express", 40, 75, 96, 97, 98, 99, 100, 76, 101, 102, 103, 104
5080 DATA 140, 77, 141, 142, 143, 144, 61, 145, 146, 66, 67, 147, 112, 148, 149, 82, 150, 151
5090 DATA 152, 153, 154, 87, 155, 156, 157, 158, 159, 160, 95
5100 DATA 11, "CG - Bklyn/Queens Crosstown", 29, 75, 96, 97, 98, 99, 100, 76, 101, 102,
5110 DATA 103, 104, 140, 77, 131, 132, 133, 134, 135, 136, 137, 138, 139, 262, 263, 264, 33,

```



```

5120 DATA 109, 110, 111
5130 '
5140 'Subroutine to check for lunch and end of workday
5150 IF MIN > TM THEN 5220 : 'After 5 pm?
5160 GOSUB 5250
5170 IF LUN = 1 THEN RETURN : 'Had lunch already?
5180 IF MIN < 180 THEN RETURN : 'Before 12 noon?
5190 IF PERS = 1 THEN RETURN : 'On a train?
5200 PRINT : PRINT "Time for a lunch break. Chili dog and cola. Burp!"
5210 PRINT : MIN = MIN + INT(24 + 20 * RND(1)) : LUN = 1 : RETURN
5220 PRINT : PRINT "So sorry, it is after " TM$ "pm and the places to which"
5230 PRINT "you want to go will be closed." : GOTO 1520
5240 '
5250 'Subroutine to print the time
5260 HR = INT(MIN / 60) : MN = MIN - 60 * HR : IF HR < 4 THEN HP = 9 + HR ELSE HP = HR - 3
5270 HP = 100 * HP + MN + 10000 : H$ = STR$(HP)
5280 PRINT : PRINT "Time " MID$(H$, 3, 2) ":" RIGHT$(H$, 2) : RETURN
5290 '
5300 'Subroutine to read yes/no answer
5310 IF A$ = "" THEN A$ = "Y" : RETURN
5320 GOSUB 5350 : IF A$ = "Y" OR A$ = "N" THEN RETURN
5330 INPUT "Don't understand your answer. Enter 'Y' or 'N' please";A$ : GOTO 5320
5340 '
5350 'Subroutine to read first letter of answer
5360 A$ = LEFT$(A$, 1) : IF A$ >= "A" AND A$ <= "Z" THEN RETURN
5370 A$ = CHR$(ASC(A$) - 32) : RETURN
5380 '
5390 'Subroutine to print delivery/pick-up log
5400 PRINT : PRINT TAB(20) "Delivery - Pick-up Log" : PRINT
5410 FOR I = 1 TO 15
5420 IF DORP(I) = 0 OR DORP(I) = 3 THEN 5450
5430 IF DORP(I) = 1 THEN X$ = "Delivery" ELSE X$ = "Pick-up"
5440 PRINT I, X$, PKGDES$(I)
5450 NEXT I : PRINT : GOSUB 5500 : RETURN
5460 '
5470 'Subroutine to make a short pause
5480 FOR I = 1 TO 1200 : NEXT I : RETURN
5490 '
5500 X$ = "Press any key to continue." : GOSUB 5530
5510 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND : PRINT : RETURN
5520 '
5530 'Subroutine to print centered lines
5540 PRINT TAB((70 - LEN(X$)) / 2) X$; : RETURN
5550 '
5560 'Subroutine to shuffle the list of packages
5570 FOR I = 1 TO PS : DUM(I) = I : NEXT I
5580 FOR I = 1 TO PS - 1
5590 K = I + INT((PS + 1 - I) * RND(1)) : J = PN(I) : PN(I) = PN(K) : PN(K) = J
5600 X$ = PKGDES$(I) : PKGDES$(I) = PKGDES$(K) : PKGDES$(K) = X$
5610 A = PKSTNU(I) : PKSTNU(I) = PKSTNU(K) : PKSTNU(K) = A
5620 FOR J = 1 TO 3
5630 A = PKGSTA(I, J) : PKGSTA(I, J) = PKGSTA(K, J) : PKGSTA(K, J) = A
5640 A = PKSTDS(I, J) : PKSTDS(I, J) = PKSTDS(K, J) : PKSTDS(K, J) = A
5650 NEXT J : NEXT I : RETURN
5660 '
5670 'Subroutine to play a short melody
5680 SOUND 262, 4 : SOUND 294, 6 : SOUND 330, 4 : SOUND 349, 6
5690 SOUND 392, 6 : SOUND 349, 10 : SOUND 330, 12 : FOR K = 1 TO 1200 : NEXT
5700 SOUND 330, 4 : FOR K = 1 TO 4 : SOUND 294, 4 : NEXT
5710 SOUND 330, 6 : SOUND 330, 6 : SOUND 262, 12 : RETURN

```

[Return to Table of Contents](#) |
 [Previous Section](#) |
 [Next Section](#)



THE GAME

In Hong Kong Hustle, you are the Tai Pan of the largest international trading house in Hong Kong. It is June 30, 1997, and China is due to take over the Colony on July 1. You have just learned that the Chinese intend to confiscate much of the property of the major trading houses. Your objective is to collect 15 bags of gold and jewels, which you have stored around the city, get them to your boat before midnight, and escape.

You disguise yourself as a factory worker and set out from Repulse Bay, using only public transportation, to visit the locations in your notebook. Here are some hints to help you in your quest to remain the wealthiest trader in Southeast Asia:

- At the beginning of the game, your boat is docked near the Macau Jetfoil Pier. You can safely move it once to another pier near Tsuen Wan in the New Territories, but to move it more than once would be very dangerous.
- You may encounter members of the Green Triad, Hong Kong's organized-crime faction. For the most part, they are interested only in money, but it is best to avoid them.
- You will find the game extremely difficult until you develop a map of the transit routes of Hong Kong. If you have a map the colony, you can navigate more easily. If you don't, you can develop your own by riding the various buses, ferries, and trains from one end of the line to the other.
- You will find that most, but not all, of your destinations can be reached by more than one transit line. Generally, travel on the MTR (Mass Transit Railway) and the Canton/Kowloon Railway is faster than travel on buses and trams.

- You will probably want to complete all pickups on Hong Kong Island before moving your boat to Tsuen Wan, which is a better base of operations in the New Territories. Pickups in Kowloon can be made equally well from either pier.

Hong Kong Hustle is, perhaps, the most challenging game in this book. Completing the assignment, therefore, is especially rewarding. And as you ride the transit system you can be planning how to spend all that gold—if you succeed. Good joss!

THE STORY

Hong Kong Island was a small uninhabited rocky island at the mouth of the Pearl River until the 1830s when, during the days of the tea and opium trade, British sea captains discovered its magnificent protected harbor. Its name derives from two Chinese characters "Heung Kong" meaning fragrant harbor. Some historians suggest that the name might have been inspired by the scent of opium-laden ships in the harbor. Others think the name may have been a reference to the incense and joss stick industries on the south and west coasts of the island. The origin of the name, however, was unimportant to the British who settled on the island and made it an important port for all Far East trade.

During the Nineteenth Century, Britain concluded three treaties with China relating to Hong Kong: the Treaty of Nanking, signed in 1842 and ratified in 1843 under which Hong Kong Island was ceded in perpetuity; the Convention of Peking in 1860 under which the southern part of the Kowloon Peninsula and Stonecutters Island were ceded in perpetuity; and the Convention of 1898, under which the New Territories and outlying islands (comprising 92% of the total land area of the colony) were leased to Britain for 99 years from July 1, 1898.

Hong Kong and the Kowloon Peninsula are relatively small—35 square miles—while the New Territories and outlying islands (235 of them) comprise a much larger land area—377 square miles. Although the British settled Hong Kong, today they represent just 2% of the population, with most of the remaining 5.3 million residents being Chinese.

It was the fact that the New Territories are subject to a lease that caused the current British Government to enter into negotiations on the future of Hong Kong with the People's Republic of China in the late 1970s. The Chinese Government has consistently taken the view that the whole of Hong Kong is Chinese territory. Its position for many years was that the question of Hong Kong fell into the category of unequal treaties of the past; that it should be settled peacefully when conditions were right; and that pending a settlement, the status quo should be maintained.

Of course, subtle hints were dropped from time to time. All of Hong Kong's water and a substantial amount of its food come from China and the New Territories, and it would be difficult for the residents of the island to exist without them. But, officially, the position was that the matter should be settled through diplomatic channels.

In the late 1970s, as the duration of the New Territories lease decreased, concern about the future of Hong Kong began to be expressed both in the colony itself and among foreign investors. In particular, there was increasing concern about individual land leases granted in the New Territories, all of which were set to expire three days before the expiration of the New Territories lease in 1997. It was clear that the short duration of these leases and the inability of the Hong Kong Government to grant new ones extending beyond 1997 would deter future investment and undermine confidence.

The British Government determined that the situation would become untenable by the mid-1980s if nothing was done. Thus, the Governor of Hong Kong visited Peking in March 1979 and attempted to solve the specific question of leases in the New Territories. He was not successful and not much happened for another two years. Then, finally, in 1982 Chinese officials indicated to a visiting British delegation that they were prepared to enter

into negotiations over Hong Kong. Accordingly, Prime Minister Margaret Thatcher met with Chairman Deng Xiaoping on September 24, 1982, and jointly they "agreed to enter into talks through diplomatic channels with the common aim of maintaining the stability and prosperity of Hong Kong." During these negotiations, Hong Kong and British delegates explained in detail the legal and commercial systems which prevail in Hong Kong and the importance to these systems of the British role and link. Following extensive discussion, however, it became clear that the continuation of British administration in any form after 1997 would be unacceptable to the Chinese.

The British then agreed to explore the possibility of a bilateral arrangement coupled with a high degree of autonomy under Chinese sovereignty to ensure lasting stability and prosperity for Hong Kong. Discussions intensified after April 1984, resulting in a Joint Declaration issued on September 26, 1984.

The declaration formally states that China will resume the exercise of sovereignty over Hong Kong Island, Kowloon, and the New Territories as of July 1, 1997. Hong Kong will be established as a "Special Administrative Region" called Hong Kong, China, which will be under the direct authority of the Central People's Government of the People's Republic of China. The Hong Kong Special Administrative Region will be vested with executive, legislative, and independent judicial power, and laws currently in force in Hong Kong will remain basically unchanged.

The government of Hong Kong will consist of officials nominated by the Central People's Government. Local Hong Kong inhabitants, British, and other foreign nationals may be employed as advisors and even hold certain (unspecified) posts in various government departments.

Because free trade has made Hong Kong what it is today, it will retain its status as a free port and a separate customs territory. Nor will it be taxed by China. In the interim, the British will continue to govern Hong Kong with the important provision that land leases can be written through the year 2047.

Reaction to the pact is mixed. Most people in Hong Kong have no other place to go and, even if they did, would have difficulty getting a permanent visa. Hence, they tend to think that Hong Kong should accept the agreement and try to make the most of it. Some British managers and administrators in Hong Kong are quietly making contingency plans, but many intend to stay and give it a try. Hong Kong-born Chinese in Britain and other parts of the world have been quick to condemn Britain's negotiations as a "giveaway" and a "farce." As one said, "Hong Kong people are trying to convince themselves that the agreement will bind China, but it's just not true. China will do what she likes as she always has in the past." Another expressed concern that "there is nothing to prevent Hong Kong from being ruled by the mainland."

Chinese officials, of course, declare that the people of Hong Kong have nothing to fear. On October 3, 1984, Deng Xiaoping told a group of visiting Hong Kong delegates, "The Chinese Government will not change its policy, nor can anyone else change it." He went on, "We have kept our word in external affairs even during periods of chaos. It is the tradition of the Chinese nation to honor its commitments."

All of which sounds good, critics say, except that when July 2, 1997, rolls around, Hong Kong will no longer be considered an "external affair" by China. As for China honoring its commitments, skeptics urge the people of Hong Kong to ask scores of American wheat farmers what happened to their three-year contracts with that country when its harvests improved in 1985. Or to ask the managers of the Ameripex International Philatelic Exposition about China's withdrawal from the exposition on opening day to protest the participation of Taiwan.

Today, Hong Kong is still a thriving, cosmopolitan city of many contrasts—elegant shops and crowded street markets, first-class restaurants and outdoor food stalls, luxury hotels and teeming tenements, bustling factories and peaceful duck farms, noisy discos and ancient temples. After Tokyo, it is the main commercial hub of the Far East, and, as the Tourist Association brochure proclaims, it is "one of life's great adventures."

BIBLIOGRAPHY

China Daily (Sept. 27, 1984, Oct. 4, 1984), Beijing, China.

Hong Kong Discovery Guide. Hong Kong Tourist Association, Connaught Centre, Hong Kong. 1986.

Hong Kong Standard (Sept. 27, 28, 29, 1984), Kowloon, Hong Kong.

Places of Interest by Public Transport, Hong Kong Tourist Association. 1985.

THE PROGRAM

The Hong Kong Hustle program is built upon the same basic framework as Subway Scavenger. It has one important addition, namely, allowing for variable-length journey segments. In Subway Scavenger, each trip on a train was assumed to last only two or three minutes.

However, it was not feasible to use such a narrow time range for journeys in the Hong Kong simulation, because some trips could be as short as two minutes (tram, MTR), while others could be as long as 20 minutes (ferry, bus over the mountains).

Another less important change was incorporated in Hong Kong Hustle, specifically the option to allow the player to walk from one transit stop to another (assuming, of course, that the stops are within walking distance).

The length of time between transit stops is contained in the data statements for each transit line. Note that the first data statement for each transit line is the same as the data for the trains in the subway program: the name of the transit line, the number of stations, and the identification number of each station. A second data statement contains the frequency of service (which is not currently used in the program) followed by the time (in minutes) between stations on the line.

These data are used in the travel routine (Lines 690–740), specifically, Line 720. Note that if the vehicle is traveling in a "forward" direction (from the beginning of the line to the end), the time ($TRT(t, s)$) associated with the previous stop is used. However, if the vehicle is traveling "backwards" (from the end of the line to the beginning), then the time associated with the next stop is used.

The following example should make it easier to understand this concept:

Index	Bus Stop No.	Time to next stop
1	1	5
2	2	4
3	3	8
4	4	7
5	5	

If a bus is traveling from Stop 2 to Stop 4, for example, the elapsed time between Stop 2 to Stop 3 (four minutes) is stored as Time 2, and the time between Stop 3 to Stop 4 (eight minutes) is stored as Time 3.

Thus, the total time for the bus to go from Stop 2 to Stop 4 is Time 2 plus Time 3, or 12 minutes. On the other hand, if the bus is going from Stop 4 to Stop 2, it still takes 12 minutes, so we would not want to add Time 4 to Time 3 (15 minutes), but rather, Time 3 to Time 2.

The routine that allows walking from one transit stop to another (Lines 1130–1220) works like the one that lets you walk from a pickup location to two or more transit stops in Subway Scavenger. It, of course, also requires more data than that program requires. These data, STAWLK(s) and STAWK(s, x), are tagged onto the end of the data statement about each station, specifying station name and transit lines that stop at that station.

Hong Kong Hustle is a rather difficult game to complete successfully. There are several ways to make it easier. The easiest way is to lower the number of bags of gold and jewels to be collected.

This variable, BG, is currently set equal to 15 in Line 190. Try setting BG equal to 11 or 12, and see how you make out. Of course, if you become very proficient at the game, you can raise BG to 17 or 18 for a real challenge.

Another way to make the game easier is to extend the time deadline past midnight, say to 4:00 A.M. or 5:00 A.M. For each hour you want to extend it, add 60 to the constant 899 in Line 4630:

```
4630 IF MIN > 899 THEN 4650
```

Unfortunately, times after 12 midnight will not be displayed correctly, but if you feel ambitious, you can modify the routine to print the time (Lines 4680–4720) to allow for times later than 12 midnight.

A third way to make the game easier is to deal with packages in only Hong Kong Island (locations 1 to 9) or only Kowloon and the New Territories (locations 10–23). This is not an easy modification to make (you must both decrease the number of bags to be collected and change the limits of the FOR...NEXT loop in Line 4910 which shuffles the list of locations). In fact, it is probably easier to keep practicing with the current game than to try to modify it in this way, but the option is there for those who are especially ambitious and/or frustrated.

PROGRAM VARIABLES

A Answer to input query
A\$ Answer to input query, string
B Answer to query about station to walk to
BAG Total bags currently being carried
BG Total bags to be collected
BGMAX Maximum number of bags that can be collected
BGTOTAL Total bags aboard sampan
BGX Maximum number of bags that can be carried
DES Destination of arriving transit vehicle
FR\$(9) Alternate words for "friend"
H\$, HP, HR Hours for printing
I Temporary iteration variable
J Temporary iteration variable
K Temporary iteration variable
LGPKG(n) Logbook number (n = package no.)
MIN Total minutes of elapsed time
MN Minutes for printing
PERS Person indicator (0 = on foot, 1 = riding)
PICK(g) Pickup flag (g = logbook no.)
PKGDES\$(n) Package destination name (n = package no.)
PKGSTA(n, s) Transit stops near package destination (n = package no., s = stop c.)
PKSTOS(n, s) Time to walk from transit stop to package destination (n=package no., s = stop no.)
PKSTNU(n) Number of stations near package destination (n = package no.)
PLACE\$(s) Place name (s = transit stop)
PN(n) Package number
PS Total packages
RN Random number
STA Transit stop number
STANU(s) Number of transport lines at transit stop (s = stop)
STATR(s, t) Transit lines that stop at transit stop (s = stop no., t = transit lines)
STAWK(s, x) Identification of transit stops within walking distance (s = stop no., x = index no.)
STAWLK(s) Transit stops within walking distance of another transit stop (s = stop no.)
STNS Total transit stops
TR Transit line number
TRAIN\$(t) Name of transit line (t = transit line no.)
TRDES\$(t, 2) Transit stops at each end of line (t = line no.)
TRFREQ(t) Frequency of service on line (t = line no.)
TRNS Total transit lines
TRSTA(t, s) Transit stops on a transit line (t = transit line no., s = stop no.)
TRSTOP(t) Number of stops on transit line (t = line no.)
TRSTX Transit stop identification indicator
TRT(t, s) Time between stops (t = line no., s = stop no.)
TRX Travel direction indicator
X Temporary numeric variable
X\$ Temporary string variable
Note: "Package" refers to bag of gold or jewels to be picked up.

PROGRAM LISTINGS

[Download HONGKONG.BAS](#) (tokenized BASIC format)

```
100 CLS : KEY OFF
110 LOCATE 10, 1 : X$ = "Hong Kong Hustle, 1997" : GOSUB 5030
120 LOCATE 13, 1 : X$ = "(c) David H. Ahl, 1986" : GOSUB 5030
130 LOCATE 23, 1 : GOSUB 5000 : GOSUB 1880
140 WHILE RN > 327767 : RN = RN - 65635! : WEND : RANDOMIZE RN
150 LOCATE 24, 1 : X$ = "(Initializing data .. please be patient)" : GOSUB 5030
160 DIM PKGDES$(25), PKSTNU(25), PKGSTA(25, 5), PKSTDS(25, 5), LGPKG(25), STAWLK(130)
170 DIM PLACE$(130), STANU(130), STATR(130, 11), TRAIN$(35), PICK(20), STAWK(130, 4)
180 DIM TRSTOP(35), TRSTA(35, 25), TRDES$(35, 2), TRFREQ(35), TRT(35, 25), PN(25)
190 PS = 23 : STNS = 127 : TRNS = 34 : BG = 15 : BGMAX = BG
200 GOSUB 2100 : 'Read words
210 GOSUB 2150 : 'Read about pickups
220 GOSUB 2450 : 'Read data about stations
230 GOSUB 3820 : 'Read about transit routes
240 GOSUB 4900 : STA = 23 : 'Shuffle pickups
```



```

250 LOCATE 24, 1 : PRINT TAB(60) " "; : LOCATE 24, 1 : GOSUB 5000
260 WHILE RN > 32767 : RN = RN - 65635! : WEND : RANDOMIZE RN : CLS
270 PRINT "You may want to print or copy this screen for later reference." : PRINT
280 PRINT "Before setting out, you make a list of the various places to stop."
290 GOSUB 4830 : 'Print pickup log
300 GOSUB 4620 : PRINT "You set out from your home overlooking Repulse Bay and"
310 PRINT "make your way down to the public bus stop." : GOTO 360
320 '
330 'Arrive at place routine
340 GOSUB 4620 : 'Display the time
350 PRINT "You have arrived at " PLACE$(STA)
360 PRINT "Public transit that stops here:"
370 FOR I = 1 TO STANU(STA)
380 PRINT " " TRAIN$(STATR(STA,I)); : NEXT I : PRINT
390 IF PERS = 0 THEN 440 : 'If on foot, go to choice
400 IF STA < > TRSTA(TR, 1) AND STA < > TRSTA(TR, TRSTOP(TR)) THEN 420
410 PRINT "End of the line. You'll have to get off." : GOTO 440
420 INPUT "Do you want to get off"; A$ : GOSUB 4740
430 IF A$ = "N" THEN 690 : 'If want to stay on, go to transit travel
440 IF STA = 5 OR STA = 50 THEN GOSUB 1430 : 'Put gold in sampan?
450 PERS = 0 : PRINT "Do you want to:" : PRINT " Make a pickup (P)"
460 PRINT " Take a bus, ferry, tram, etc. (T)"
470 PRINT " Walk to another transit stop (W)"
480 PRINT " Check your log (C)"
490 INPUT "Your choice please (P, T, W, or C)"; A$
500 GOSUB 4780 : IF A$ = "P" THEN 760 ELSE IF A$ = "T" THEN 540
510 IF A$ = "W" THEN 1130 ELSE IF A$ = "C" THEN GOSUB 4830 : GOTO 450
520 INPUT "Not a valid choice, Enter P, T, W, or C please"; A$ : GOTO 500
530 '
540 'Transit arriving routine
550 GOSUB 4620 : 'Display the time (again)
560 TR = STATR(STA, INT(1 + STANU(STA) * RND(1)))
570 IF STA = TRSTA(TR, 1) THEN DES = 2 : GOTO 600 : 'At one end of line?
580 IF STA = TRSTA(TR, TRSTOP(TR)) THEN DES = 1 : GOTO 600 : 'or the other?
590 DES = INT(1 + 2 * RND(1)) : 'Destination of next transit to arrive
600 PRINT "Here comes the " TRAIN$(TR) " to " TRDES$(TR, DES)
610 MIN = MIN + 1
620 INPUT "Do you want to get on"; A$ : GOSUB 4740
630 IF A$ = "N" THEN 560 : 'If don't get on, wait for next transit
640 FOR I = 1 TO TRSTOP(TR) : 'Find out where train is
650 IF TRSTA(TR, I) < > STA THEN NEXT I : PRINT "ERROR in station location"
660 TRSTX = I : 'Transit station identification index
670 IF DES = 1 THEN TRX = -1 ELSE TRX = 1 : 'Which way are we going?
680 '
690 'Travel routine
700 PERS = 1 : PRINT "You are on the " TRAIN$(TR) " to " TRDES$(TR, DES)
710 GOSUB 1240 : 'Possible trip hazards
720 IF TRX = 1 THEN MIN = MIN + TRT(TR, TRSTX) ELSE MIN = MIN + TRT(TR, TRSTX - 1)
730 TRSTX = TRSTX + TRX : STA = TRSTA(TR, TRSTX)
740 GOTO 330 : 'Arrive at next location
750 '
760 'Pickup routine
770 IF BGX = 0 THEN 790 : 'Maximum limit not set yet?
780 IF BAG >= BGX THEN PRINT "You can't carry any more bags." : GOTO 880
790 INPUT "Which pickup do you want to make (by Logbook number)"; A
800 IF PICK(A) = 0 THEN 840
810 INPUT "That number seems to be in error. Want to check your logbook"; A$
820 GOSUB 4740 : IF A$ = "Y" THEN GOSUB 4830
830 GOTO 450 : 'Goto choice routine
840 PRINT "That pickup is at the " PKGDES$(A)
850 FOR I = 1 TO PKSTNU(A)
860 IF PKGSTA(A, I) = STA THEN 890
870 NEXT I : PRINT "...which is too far to walk from here."
880 PRINT "Perhaps you should try something else." : GOTO 450
890 PRINT "which is about a" PKSTD$(A, I) "minute walk from here. Off you go."
900 '
910 'Successful pickup
920 GOSUB 5060 : MIN = MIN + PKSTD$(A, I) + INT(5 + 10 * RND(1)) : BAG = BAG + 1 : 'Add to
time
930 PRINT "Your " FR$(INT(1 + 9 * RND(1))) " gives you the bag he has been holding"
940 PRINT "for you and wishes you good joss." : PICK(A) = 1

```

```

950 '
960 'Check to see if you are overloaded
970 IF BAG < 3 THEN 1030 : 'If carrying fewer than 3 bags then no problem
980 IF BGX = 0 THEN BGX = INT(3 + 3 * RND(1)) : 'Set maximum bags
990 IF BAG < BGX THEN 1030 : 'Haven't reached load limit yet?
1000 PRINT "That last bag was a heavy one. You can't carry any more. You'll"
1010 PRINT "have to return to your sampan and unload."
1020 '
1030 'Walk from pickup back to transit stop
1040 X = PKSTNU(A) : IF X = 1 THEN X$ = "" ELSE X$ = "s"
1050 PRINT : PRINT "From here you can walk to the following transit stop" X$ ":"
1060 IF X = 1 THEN MIN = MIN + PKSTDS(A, 1) : PRINT " " PLACE$(STA) : GOTO 330
1070 FOR I = 1 TO PKSTNU(A) : 'Iterate through nearby transit stops
1080 PRINT " " I ".. " PLACE$(PKGSTA(A, I)) : NEXT I
1090 INPUT "Which place do you want to go to (by number)";B
1100 IF B < 1 OR B > PKSTNU(A) THEN PRINT "Not a valid response." : GOTO 1090
1110 STA = PKGSTA(A, B) : MIN = MIN + PKSTDS(A, B) : GOTO 330
1120 '
1130 'Walk to another location routine
1140 X = STAWLK(STA) : IF X > 1 THEN 1160
1150 PRINT "It would take too long to walk to another transit stop." : GOTO 450
1160 PRINT : PRINT "From here you can walk to the following transit stops:"
1170 PRINT " 1 - " PLACE$(STA) " (where you are now)"
1180 FOR I = 2 TO STAWLK(STA)
1190 PRINT " " I "-" PLACE$(STAWK(STA, I)) : NEXT I
1200 INPUT "Which place do you want to go to (by number)";B
1210 IF B < 1 OR B > STAWLK(STA) THEN PRINT "Not a valid response." : GOTO 1200
1220 STA = STAWK(STA, B) : MIN = MIN + INT(6 + 6 * RND(1)) : GOTO 330
1230 '
1240 'Travel Hazards Subroutine
1250 INPUT "Do you want to put the bags you are carrying aboard your sampan";A$
1260 GOSUB 4740 : IF A$ = "N" THEN PRINT "Okay, it's up to you." : GOTO 1510
1270 PRINT "Good. You stow them safely out of sight."
1280 MIN = MIN + 8 : BGTOTAL = BGTOTAL + BAG : BAG = 0 : BGX = 0
1290 IF BGTOTAL >= BGMAX THEN 1600
1300 IF STA = 50 THEN RETURN
1310 INPUT "Do you want to move the sampan to Tsuen Wan";A$ : GOSUB 4740
1320 IF A$ = "N" THEN PRINT "Okay; the captain is ready when you are." : RETURN
1330 PRINT "Okay. You shove off and make your silent way across the harbor."
1340 MIN = MIN + INT(20 + 20 * RND(1)) : STA = 50 : GOSUB 4680
1350 PRINT "You are at Tsuen Wan, New Territories."
1360 PRINT "Public transit that stops here:"
1370 FOR I = 1 TO STANU(STA)
1380 PRINT " " TRAIN$(STATR(STA, I)); : NEXT I : PRINT : RETURN
1390 '
1400 'All gold collected before midnight?
1410 FOR I = 1 TO 3000 : NEXT I : CLS : FOR I = 1 TO 10
1420 BEEP : X$ = "CONGRATULATIONS !" : LOCATE 12, 1 : GOSUB 5030
1430 FOR K = 1 TO 100 : NEXT K : CLS : FOR K = 1 TO 50 : NEXT K : NEXT I
1440 PRINT " You managed to pick up all" BG "bags of gold and jewels"
1450 PRINT "before midnight." : PRINT
1460 IF BGTOTAL >= BG THEN 1680
1470 PRINT " It's too bad that you lost" BG - BGTOTAL "of them along the way."
1480 PRINT : PRINT " You sail away on your sampan and start your next"
1490 PRINT "great empire in Morristown, New Jersey." : PRINT
1500 X$ = "Good Joss !" : GOSUB 5030 : GOTO 1840
1510 '
1520 'Time ran out
1530 BGTOTAL = BGTOTAL + BAG
1540 IF BGTOTAL < BG THEN 1770
1550 BGTOTAL = BGTOTAL - 1 : PRINT
1560 PRINT "Too bad, in your rush to escape you had to drop a bag of gold."
1570 IF BGTOTAL < .6 * BG THEN 1800
1580 PRINT " You managed to get away with your life and" BGTOTAL "bags of"
1590 PRINT "gold and jewels. Not bad, but you could do better." : GOTO 1840
1600 PRINT " You barely managed to escape with your life and only" BGTOTAL
1610 PRINT "bags of gold and jewels. You lost much face and you will have"
1620 PRINT "difficulty becoming Tai Pan of a new venture."
1630 '
1640 'Play again routine
1650 PRINT : PRINT : INPUT "Would you like to play again";A$ : GOSUB 4740

```

```

1860 IF A$ = "Y" THEN RUN ELSE CLS : KEY ON : END
1870 '
1880 CLS : X$ = "Hong Kong Hustle" : GOSUB 5030 : PRINT : PRINT
1890 PRINT " It is June 30, 1997 and China will take over the British"
1900 PRINT "Colony of Hong Kong on July 1. While the transition was supposed"
1910 PRINT "to be smooth and amicable, you just Learned that the Chinese intend"
1920 PRINT "to confiscate much of the property of the great trading houses."
1930 PRINT " You, the Tai Pan, are being closely watched, so you disguise"
1940 PRINT "yourself as a common factory worker and set out, using only public"
1950 PRINT "transport, to recover as much of your liquid assets (gold & jewels)"
1960 PRINT "as possible before the day ends. You deposit them on an inconspic-"
1970 PRINT "uous sampan initially tied up near the Macau Jetfoil Pier."
1980 PRINT " You may move the sampan from the pier on Hong Kong Island to"
1990 PRINT "another pier near Tsuen Wan in the New Territories, but to move it"
2000 PRINT "more than once would be very dangerous."
2010 PRINT " You can use any of 34 different transit lines (bus, train,"
2020 PRINT "subway, ferry, tram, and public sampan) which service 125 stops"
2030 PRINT "throughout Hong Kong Island, Kowloon, and the New Territories. Of"
2040 PRINT "course, only 15 of these stops are of really keen interest to you."
2050 PRINT " Depending upon how much gold you pick up at various locations,"
2060 PRINT "you may have to return to your sampan more than once. Time, of"
2070 PRINT "course, is your biggest enemy, and you'll have to leave at midnight"
2080 PRINT "no matter what. Good luck!" : PRINT : RETURN
2090 '
2100 'Subroutine to read words
2110 FOR I = 1 TO 9 : READ FR$(I) : NEXT I : RETURN
2120 DATA "associate", "friend", "confidant", "ally", "comrade", "colleague", "mate"
2130 DATA "partner", "compatriot"
2140 '
2150 'Subroutine to read data about gold pickups
2160 FOR I = 1 TO PS
2170 READ PN(I), PKGDES$(I), PKSTNU(I)
2180 FOR J = 1 TO PKSTNU(I)
2190 READ PKGSTA(I, J), PKSTDS(I, J)
2200 NEXT J : NEXT I : RETURN
2210 DATA 1, "Computer Science Dept, HK University", 1, 2, 5
2220 DATA 2, "Angora Fashion Shop, Stanley", 1, 21, 5
2230 DATA 3, "Jumbo Floating Restaurant, Aberdeen", 1, 27, 10
2240 DATA 4, "Chop Shop, Peak Tower Village", 1, 32, 3
2250 DATA 5, "Cathay Hotel, Yee Woo St & Causeway Rd", 2, 11, 12, 12, 3
2260 DATA 6, "Singapore Hotel, Wan Chai District", 2, 8, 5, 9, 10
2270 DATA 7, "Owners Box 4, Happy Valley Raceway", 1, 35, 5
2280 DATA 8, "Happy Toy Company, Quarry Bay", 1, 14, 4
2290 DATA 9, "Torture Exhibit, Tiger Balm Garden", 1, 47, 7
2300 DATA 10, "Ping Shan Pagoda, New Territories", 1, 100, 6
2310 DATA 11, "Royal Hong Kong Golf Club, N.T.", 1, 106, 5
2320 DATA 12, "Village Hall at Sha Tau Kok, N.T.", 1, 86, 3
2330 DATA 13, "Harbormaster, Plover Cove, N.T.", 2, 92, 2, 91, 6
2340 DATA 14, "Tai Ping Carpet Factory, Tai Po, N.T.", 1, 81, 5
2350 DATA 15, "Temple of 1000 Buddhas, Sha Tin, N.T.", 1, 78, 10
2360 DATA 16, "Yacht Squadron Captain, Hebe Haven, N.T.", 1, 114, 3
2370 DATA 17, "Tai On Restaurant, Lyemun Village, Kowloon", 1, 127, 10
2380 DATA 18, "Kam Tin Walled Village, N.T.", 1, 97, 3
2390 DATA 19, "San Miguel Brewery, Ting Kau, N.T.", 1, 118, 5
2400 DATA 20, "Lei Cheng Uk Tomb, Cheung Sha Wan, Kowloon", 1, 57, 4
2410 DATA 21, "Sung Dynasty Village, Lai Chi Kok, Kowloon", 1, 58, 3
2420 DATA 22, "Wong Tai Sin Temple, Kowloon", 1, 66, 5
2430 DATA 23, "Jade Market near Jordan Rd, Kowloon", 2, 49, 3, 52, 3
2440 '
2450 'Subroutine to read data about transit stops
2460 FOR I = 1 TO STNS : 'STNS = number of stations
2470 READ STA, PLACE$(I), STANU(I), STAWLK(I)
2480 FOR J = 1 TO STANU(I)
2490 READ STATR(I, J) : 'Read trains, buses, etc. that stop here
2500 NEXT J
2510 FOR J = 1 TO STAWLK(I)
2520 READ STAWK(I, J) : 'Read other stations within walking distance
2530 NEXT J : NEXT I : RETURN
2540 DATA 1, "Whitty Street", 2, 1, 4, 8, 1
2550 DATA 2, "Hong Kong University", 2, 1, 4, 8, 2
2560 DATA 3, "Central Market", 3, 1, 4, 8, 19, 3

```

2570 DATA 4, "Sheung Wan District", 3, 1, 4, 8, 19, 4
2580 DATA 5, "Macau Jet Foil Pier", 5, 3, 1, 2, 4, 8, 19, 5, 6, 46
2590 DATA 6, "Central Bus Terminal", 9, 4, 4, 5, 6, 7, 8, 10, 11, 15, 19, 5, 5, 43, 46
2600 DATA 7, "Queensway/Admiralty", 11, 3, 1, 2, 4, 5, 6, 10, 11, 15, 16, 17, 19, 7, 8, 37
2610 DATA 8, "Arsenal Street", 7, 3, 1, 2, 4, 5, 6, 11, 19, 8, 7, 37
2620 DATA 9, "Wan Chai Road", 3, 2, 2, 4, 11, 9, 10
2630 DATA 10, "Tin Lok Lane and Hennessy Road", 4, 3, 2, 4, 11, 19, 10, 9, 11
2640 DATA 11, "Percival and Yee Woo Streets", 4, 3, 2, 4, 11, 16, 11, 10, 12
2650 DATA 12, "Causeway Road (Causeway Bay)", 3, 2, 2, 4, 16, 12, 11
2660 DATA 13, "North Point", 3, 1, 2, 4, 16, 13
2670 DATA 14, "Quarry Bay", 3, 1, 2, 4, 16, 14
2680 DATA 15, "Sai Wan Ho Ferry Pier", 5, 2, 2, 4, 14, 16, 33, 15, 16
2690 DATA 16, "Shau Kei Wan", 4, 2, 2, 4, 14, 16, 16, 15
2700 DATA 17, "Mt. Parker", 1, 1, 14, 17
2710 DATA 18, "Shek-O Road", 1, 1, 14, 18
2720 DATA 19, "Turtle Cove Beach", 1, 1, 14, 19
2730 DATA 20, "Hairpin Beach", 1, 1, 14, 20
2740 DATA 21, "Stanley", 4, 1, 6, 7, 9, 14, 21
2750 DATA 22, "Chung Horn Wan Road", 2, 1, 6, 9, 22
2760 DATA 23, "Repulse Bay", 3, 1, 5, 6, 9, 23
2770 DATA 24, "Deep Water Bay", 1, 1, 9, 24
2780 DATA 25, "Wong Chuk Hang", 1, 1, 9, 25
2790 DATA 26, "Ocean Park", 2, 1, 9, 10, 26
2800 DATA 27, "Aberdeen", 3, 1, 8, 9, 10, 27
2810 DATA 28, "Wah Fu", 1, 1, 8, 28
2820 DATA 29, "Pok Fu Lam", 1, 1, 8, 29
2830 DATA 30, "Mt. Davis", 1, 1, 8, 30
2840 DATA 31, "Kennedy Town", 1, 1, 8, 31
2850 DATA 32, "Victoria Peak", 2, 1, 3, 15, 32
2860 DATA 33, "Magazine Gap", 1, 1, 10, 33
2870 DATA 34, "Aberdeen Tunnel", 1, 1, 10, 34
2880 DATA 35, "Happy Valley Raceway", 2, 1, 1, 19, 35
2890 DATA 36, "Garden Street", 1, 3, 3, 36, 7, 46
2900 DATA 37, "Queens Road East", 8, 3, 1, 2, 4, 5, 6, 7, 11, 19, 37, 7, 8
2910 DATA 38, "Mt. Nicholson", 2, 1, 5, 6, 38
2920 DATA 39, "Wong Nei Chong Gap Road", 2, 1, 5, 6, 39
2930 DATA 40, "Tai Koo Shing", 3, 1, 2, 4, 16, 40
2940 DATA 41, "Fortress Hill", 3, 1, 2, 4, 16, 41
2950 DATA 42, "Tin Hau", 3, 1, 2, 4, 16, 42
2960 DATA 43, "Star Ferry Pier, Hong Kong", 4, 3, 1, 2, 12, 13, 43, 6, 46
2970 DATA 44, "Heng Fa Chuen", 1, 1, 16, 44
2980 DATA 45, "Chai Wan", 1, 1, 16, 45
2990 DATA 46, "Central", 9, 4, 4, 5, 6, 10, 11, 15, 16, 17, 19, 46, 6, 36, 43
3000 DATA 47, "Tiger Balm Garden", 1, 1, 11, 47
3010 DATA 48, "Star Ferry Pier, Kowloon", 6, 3, 12, 20, 21, 22, 28, 31, 48, 51, 107
3020 DATA 49, "Jordan Road Ferry Pier", 3, 2, 13, 23, 24, 49, 52
3030 DATA 50, "Tsuen Wan", 3, 1, 17, 22, 23, 50
3040 DATA 51, "Tsim Sha Tsui, Haiphong & Nathan Rds", 4, 3, 17, 21, 22, 31, 51, 48, 107
3050 DATA 52, "Jordan Rd & Nathan Rd", 4, 3, 17, 21, 22, 31, 52, 49, 53
3060 DATA 53, "Yau Ma Tei & Nathan Rd", 6, 3, 17, 18, 21, 23, 24, 31, 53, 52, 54
3070 DATA 54, "Mong Kok", 6, 3, 17, 18, 21, 22, 24, 31, 54, 53, 55
3080 DATA 55, "Prince Edward", 5, 2, 17, 18, 21, 24, 31, 55, 54
3090 DATA 56, "Sham Shui Po", 1, 1, 17, 56
3100 DATA 57, "Cheung Sha Wan", 1, 1, 17, 57
3110 DATA 58, "Lai Chi Kok", 4, 1, 17, 21, 22, 23, 58
3120 DATA 59, "Mei Foo", 3, 1, 17, 22, 23, 59
3130 DATA 60, "Lai King", 3, 1, 17, 22, 23, 60
3140 DATA 61, "Kwai Fong", 3, 1, 17, 22, 23, 61
3150 DATA 62, "Kwai Hing", 3, 1, 17, 22, 23, 62
3160 DATA 63, "Tai Wo Hau", 3, 1, 17, 22, 23, 63
3170 DATA 64, "Shek Kip Mei", 5, 1, 18, 21, 22, 24, 31, 64
3180 DATA 65, "Lok Fu", 1, 1, 18, 65
3190 DATA 66, "Wong Tai Sin", 1, 1, 18, 66
3200 DATA 67, "Diamond Hill", 1, 1, 18, 67
3210 DATA 68, "Choi Hung Terminal", 3, 1, 18, 28, 29, 68
3220 DATA 69, "Kowloon Bay", 1, 1, 18, 69
3230 DATA 70, "Ngau Tau Kong", 1, 1, 18, 70
3240 DATA 71, "Kwun Tong", 2, 1, 18, 34, 71
3250 DATA 72, "Tai Wo Ping", 1, 1, 31, 72
3260 DATA 73, "Tsim Sha Tsui Centre", 2, 2, 20, 28, 73, 107
3270 DATA 74, "Hung Hom Train Station", 2, 1, 20, 30, 74

```

3280 DATA 75, "Mong Kok", 1, 1, 30, 75
3290 DATA 76, "Kowloon Tong", 3, 1, 18, 24, 30, 76
3300 DATA 77, "Tai Wai", 1, 1, 30, 77
3310 DATA 78, "Sha Tin", 2, 1, 24, 30, 78
3320 DATA 79, "Fo Tan (Sha Tin New Town)", 2, 1, 24, 30, 79
3330 DATA 80, "Chinese University", 2, 1, 24, 30, 80
3340 DATA 81, "Tai Po", 4, 1, 24, 25, 27, 30, 81
3350 DATA 82, "Fanling", 3, 2, 24, 26, 30, 82, 84
3360 DATA 83, "Sheung Shui", 3, 1, 24, 30, 32, 83
3370 DATA 84, "Luen Wo Market", 1, 2, 26, 84, 82
3380 DATA 85, "Hung Leng", 1, 1, 26, 85
3390 DATA 86, "Sha Tau Kok", 1, 1, 26, 86
3400 DATA 87, "Starling Inlet", 1, 1, 26, 87
3410 DATA 88, "Luk Keng", 1, 1, 26, 88
3420 DATA 89, "a small crossroad with a sign to Tin Sam", 1, 1, 27, 89
3430 DATA 90, "Shuen Wan", 1, 1, 27, 90
3440 DATA 91, "Tolo Harbor", 1, 2, 27, 91, 92
3450 DATA 92, "Plover Cove Dock", 1, 2, 27, 92, 91
3460 DATA 93, "Tai Mei Tuk", 1, 1, 27, 93
3470 DATA 94, "Brides's Pool", 1, 1, 27, 94
3480 DATA 95, "a rutted crossroad with a dirty sign, 'Ping Long'", 1, 1, 25, 95
3490 DATA 96, "a mountain pass by Route Twisk Lookout", 1, 1, 25, 96
3500 DATA 97, "Kam Tin Walled Village", 1, 1, 25, 97
3510 DATA 98, "a chicken farm near Ho Hok Shan", 1, 1, 25, 98
3520 DATA 99, "Yuen Long", 3, 1, 23, 25, 32, 99
3530 DATA 100, "Ping Shan Pagoda", 1, 1, 25, 100
3540 DATA 101, "Lau Fau Shan", 1, 1, 25, 101
3550 DATA 102, "Puk Wai", 1, 1, 32, 102
3560 DATA 103, "Mai Po", 1, 1, 32, 103
3570 DATA 104, "Lok Ma Chau", 1, 1, 32, 104
3580 DATA 105, "a duck farm near Pak Shek", 1, 1, 32, 105
3590 DATA 106, "Royal HK Golf Club", 1, 1, 32, 106
3600 DATA 107, "Salisbury Rd", 2, 4, 20, 28, 107, 48, 51, 73
3610 DATA 108, "Chatham Rd", 1, 1, 28, 108
3620 DATA 109, "Kai Tok Airport", 1, 1, 28, 109
3630 DATA 110, "Ngau Chi Wan", 1, 1, 28, 110
3640 DATA 111, "Tseng Lan Shue", 1, 1, 29, 111
3650 DATA 112, "Pik Uk", 1, 1, 29, 112
3660 DATA 113, "Wo Mei", 1, 1, 29, 113
3670 DATA 114, "Marina Cove, Hebe Haven", 1, 1, 29, 114
3680 DATA 115, "Ta Ho Tun", 1, 1, 29, 115
3690 DATA 116, "Sai Kung Town", 1, 1, 29, 116
3700 DATA 117, "Lucky Factory Estate, Tso Kung Tam", 1, 1, 23, 117
3710 DATA 118, "San Miguel Brewery, Ting Kau", 1, 1, 23, 118
3720 DATA 119, "Dragon Beach, Tai Lam", 1, 1, 23, 119
3730 DATA 120, "Tuen Mun-Castle Peak", 1, 1, 23, 120
3740 DATA 121, "Ching Chung Koon", 1, 1, 23, 121
3750 DATA 122, "Miu Fat Monastery", 1, 1, 23, 122
3760 DATA 123, "Lion Rock Tunnel", 1, 1, 24, 123
3770 DATA 124, "Amah Rock", 1, 1, 24, 124
3780 DATA 125, "Kak Tin", 1, 1, 24, 125
3790 DATA 126, "Tsang Tai Uk", 1, 1, 24, 126
3800 DATA 127, "Lyemun", 2, 1, 33, 34, 127
3810 '
3820 'Subroutine to read data about transit routes
3830 FOR I = 1 TO TRNS : 'TRNS = number of trains, buses, etc.
3840 READ TR, TRAIN$(I), TRSTOP(I)
3850 FOR J = 1 TO TRSTOP(I)
3860 READ TRSTA(I, J) : 'Read station numbers for transit vehicle
3870 NEXT J : READ TRFREQ(I)
3880 FOR J = 1 TO TRSTOP(I) - 1 : 'Read travel times
3890 READ TRT(I, J) : NEXT J
3900 TRDES$(TR, 1) = PLACE$(TRSTA(TR, 1)) : 'Place name at one end of line
3910 TRDES$(TR, 2) = PLACE$(TRSTA(TR, TRSTOP(TR))) : 'and at other end
3920 NEXT I : RETURN
3930 DATA 1, "#1 Bus", 6, 5, 43, 7, 37, 8, 35
3940 DATA 5, 3, 3, 3, 2, 9
3950 DATA 2, "#2 Bus", 16, 5, 43, 7, 37, 8, 9, 10, 11, 12, 42, 41, 13, 14, 40, 15, 16
3960 DATA 4, 3, 3, 2, 2, 3, 3, 2, 3, 3, 3, 3, 3, 3, 2
3970 DATA 3, "Peak Tram", 2, 32, 36
3980 DATA 12, 4

```

```

3990 DATA 4, "Tram A", 21, 1, 2, 3, 4, 5, 6, 46, 7, 37, 8, 9, 10, 11, 12, 42, 41, 13, 14, 40,
15, 16
4000 DATA 4, 4, 3, 3, 3, 2, 2, 2, 2, 3, 2, 3, 3, 2, 3, 4, 4, 4, 4, 3
4010 DATA 5, "#61 Bus", 8, 6, 46, 7, 37, 8, 38, 39, 23
4020 DATA 20, 2, 3, 3, 2, 4, 4, 4
4030 DATA 6, "#6 Bus", 10, 6, 46, 7, 37, 8, 38, 39, 23, 22, 21
4040 DATA 12, 2, 3, 3, 2, 4, 4, 4, 5, 5
4050 DATA 7, "#260 Express Bus", 3, 6, 37, 21
4060 DATA 25, 6, 19
4070 DATA 8, "#7 Bus", 11, 6, 5, 4, 3, 2, 1, 31, 30, 29, 28, 27
4080 DATA 5, 2, 3, 3, 2, 3, 3, 4, 4, 3, 3
4090 DATA 9, "#73 Bus", 7, 21, 22, 23, 24, 25, 26, 27
4100 DATA 15, 5, 5, 7, 8, 8, 7
4110 DATA 10, "#70 Bus", 7, 6, 46, 7, 33, 34, 26, 27
4120 DATA 10, 2, 3, 3, 4, 3, 7
4130 DATA 11, "#11 Bus", 9, 6, 46, 7, 37, 8, 9, 10, 11, 47
4140 DATA 7, 2, 3, 3, 2, 2, 3, 3, 9
4150 DATA 12, "Star Ferry", 2, 43, 48
4160 DATA 6, 12
4170 DATA 13, "Jordan Road Ferry", 2, 43, 49
4180 DATA 12, 16
4190 DATA 14, "#14 Bus", 7, 15, 16, 17, 18, 19, 20, 21
4200 DATA 45, 3, 5, 6, 6, 5, 5
4210 DATA 15, "#15 Bus", 4, 6, 46, 7, 32
4220 DATA 17, 2, 3, 8
4230 DATA 16, "MTR-Red", 13, 46, 7, 11, 12, 42, 41, 13, 14, 40, 15, 16, 44, 45
4240 DATA 8, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
4250 DATA 17, "MTR-Green", 16, 46, 7, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 50
4260 DATA 3, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
4270 DATA 18, "MTR-Blue", 12, 53, 54, 55, 64, 76, 65, 66, 67, 68, 69, 70, 71
4280 DATA 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
4290 DATA 19, "Tram B", 10, 3, 4, 5, 6, 46, 7, 37, 8, 10, 35
4300 DATA 4, 3, 3, 2, 2, 2, 2, 2, 3, 2, 8
4310 DATA 20, "Mini Bus", 4, 48, 107, 73, 74
4320 DATA 3, 2, 2, 3
4330 DATA 21, "#6A Bus", 8, 48, 51, 52, 53, 54, 55, 64, 58
4340 DATA 3, 2, 2, 3, 2, 2, 3, 4
4350 DATA 22, "#2 Bus", 12, 48, 51, 52, 54, 64, 58, 59, 60, 61, 62, 63, 50
4360 DATA 3, 2, 2, 4, 3, 4, 2, 3, 3, 3, 3, 3
4370 DATA 23, "#68 Bus", 17, 49, 53, 64, 58, 59, 60, 61, 62, 63, 50, 117, 118, 119, 120, 121,
122
4380 DATA 99, 5, 2, 3, 4, 2, 3, 3, 3, 3, 3, 5, 7, 6, 8, 6, 8, 6
4390 DATA 24, "#70 Bus", 16, 49, 53, 54, 55, 64, 76, 123, 124, 125, 126, 78, 79, 80, 81, 82,
83
4400 DATA 5, 2, 2, 2, 4, 4, 5, 4, 5, 4, 5, 6, 5, 6, 5, 4
4410 DATA 25, "#64K Bus", 8, 81, 95, 96, 97, 98, 99, 100, 101
4420 DATA 10, 5, 6, 5, 6, 5, 6, 6
4430 DATA 26, "#69K Bus", 6, 82, 84, 85, 86, 87, 88
4440 DATA 12, 2, 8, 5, 6, 5
4450 DATA 27, "#74R Bus", 7, 81, 89, 90, 91, 92, 93, 94
4460 DATA 15, 5, 5, 6, 3, 8, 10
4470 DATA 28, "#5 Bus", 6, 48, 107, 108, 109, 110, 68
4480 DATA 6, 2, 3, 4, 4, 3
4490 DATA 29, "#92 Bus", 7, 68, 111, 112, 113, 114, 115, 116
4500 DATA 9, 3, 4, 3, 4, 3, 4
4510 DATA 30, "Kowloon-Canton Railway", 10, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83
4520 DATA 8, 3, 3, 3, 4, 3, 3, 4, 4, 4
4530 DATA 31, "#9 Bus", 8, 48, 51, 52, 53, 54, 55, 64, 72
4540 DATA 5, 2, 2, 2, 3, 2, 3, 3
4550 DATA 32, "#76K Bus", 7, 99, 102, 103, 104, 105, 106, 83
4560 DATA 15, 6, 6, 5, 6, 6, 6
4570 DATA 33, "HYF Ferry", 2, 15, 127
4580 DATA 15, 12
4590 DATA 34, "Sampan", 2, 71, 127
4600 DATA 30, 10
4610 '
4620 'Subroutine to check for out of time
4630 IF MIN > 899 THEN 4650 : '12 midnight yet?
4640 GOSUB 4680 : RETURN
4650 PRINT : PRINT "So sorry, it is after 12 midnight and you'll have to get to"
4660 PRINT "your sampan and out of Hong Kong as quickly as possible." : GOTO 1720

```

```

4670 '
4680 'Subroutine to print the time
4690 HR = INT(MIN / 60) : MN = MIN - 60 * HR : IF HR < 4 THEN HP = 9 + HR ELSE HP = HR - 3
4700 IF MIN < 181 OR MIN > 900 THEN X$ = "a.m." ELSE X$ = "p.m."
4710 HP = 100 * HP + MN + 10000 : H$ = STR$(HP)
4720 PRINT : PRINT "Time " MID$(H$, 3, 2) " : " RIGHT$(H$, 2) " " X$ : RETURN
4730 '
4740 'Subroutine to read yes/no answer
4750 GOSUB 4790 : IF A$ = "Y" OR A$ = "N" THEN RETURN
4760 INPUT "Don't understand your answer. Enter 'Y' or 'N' please";A$ : GOTO 4750
4770 '
4780 'Subroutine to read first letter of answer
4790 IF A$ = "" THEN A$ = "Y" : RETURN
4800 A$ = LEFT$(A$, 1) : IF A$ >= "A" AND A$ <= "Z" THEN RETURN
4810 A$ = CHR$(ASC(A$) - 32) : RETURN
4820 '
4830 'Subroutine to print pickup log
4840 PRINT : PRINT "Your pickup notebook shows:"
4850 FOR I = 1 TO BG
4860 IF PICK(I) = 1 THEN 4880
4870 PRINT I, PKGDES$(I)
4880 NEXT I : PRINT : GOSUB 5000 : RETURN
4890 '
4900 'Subroutine to shuffle the list of pickups
4910 FOR I = 1 TO PS - 1
4920 K = I + INT((PS + 1 - I) * RND(1)) : J = PN(I) : PN(I) = PN(K) : PN(K) = J
4930 X$ = PKGDES$(I) : PKGDES$(I) = PKGDES$(K) : PKGDES$(K) = X$
4940 A = PKSTNU(I) : PKSTNU(I) = PKSTNU(K) : PKSTNU(K) = A
4950 FOR J = 1 TO 4
4960 A = PKGSTA(I, J) : PKGSTA(I, J) = PKGSTA(K, J) : PKGSTA(K, J) = A
4970 A = PKSTDS(I, J) : PKSTDS(I, J) = PKSTDS(K, J) : PKSTDS(K, J) = A
4980 NEXT J : NEXT I : RETURN
4990 '
5000 X$ = "Press any key to continue." : GOSUB 5030
5010 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND : PRINT : RETURN
5020 '
5030 'Subroutine to print centered lines
5040 PRINT TAB((70 - LEN(X$)) / 2) X$; : RETURN
5050 '
5060 'Short delay
5070 FOR J = 1 TO 2500 : NEXT J : RETURN

```

[Return to Table of Contents](#) |
 [Previous Section](#) |
 [Next Section](#)



THE GAME

In Voyage to Neptune, you are in command of the first manned spaceship journeying from Earth to Neptune. The year is 2100, and manned space stations have been established at roughly 500-million-mile intervals along your route. These stations are at Jupiter (orbiting Callisto), Saturn (orbiting Titan), Uranus (on Ariel), and at several points midway between Saturn and Uranus, and Uranus and Neptune.

Navigation of your spacecraft is highly computerized, and you have a competent personnel officer on board to handle the inevitable people problems that will arise during a six-year journey. Your primary function as captain, therefore, is to maintain the delicate balance that will provide your ship with enough energy to reach its destination and return to Earth.

The following hints will help to speed you on your way:

- Your spaceship uses both solar energy and nuclear fuel. Because you cannot carry enough nuclear fuel for the entire trip, you also have on board a multi-celled breeder reactor, which takes the spent fuel from the propulsion engines along with a small amount of primary fuel and turns it—somewhat inefficiently—back into primary fuel for propulsion. You must decide how much fuel to use for propulsion and how many breeder reactor cells to operate on each leg of the trip.
- At each space station along the way, you may trade nuclear fuel for breeder-reactor cells and vice versa. You may not trade more fuel than you have; nor may you trade so many cells that your breeder reactor ceases to function.
- You may have problems with your engines along the way, but there is little you can do to lessen this possibility. Likewise, if you are unlucky, some of your fuel may decay.

If all goes well and you make judicious use of your resources, you can reach Neptune in just under six years with plenty of fuel left for your return voyage. Can you do it the first time or will you need a few trial runs?

THE STORY

Although fictional accounts of space voyages date back to the 1500s, the first serious rocket motor for space flight was not devised until the twentieth century. In 1903, K.E. Tsiolkovsky, a Russian pioneer of astronautics, proposed using a liquid-propelled rocket motor for space flight. The first rocket to employ an engine of this type was fired by Robert H. Goddard in the United States in 1926.

In Germany during World War II, Werner von Braun developed the V2 weapon, the first long-range liquid-fuel rocket. After the war, von Braun and most of the other German rocket specialists went to the United States, where they were instrumental in the development of rockets for satellite launches and space exploration.

Russia was also working on rockets and space vehicles and, on October 4, 1957, launched Sputnik I, the event that marked the real beginning of the space age. Other Soviet satellites followed and in 1958 the United States launched its first satellite, Explorer I, which provided the first information about the Van Allen radiation belts that surround the Earth.

In January 1959, the Soviet satellite Luna I was the first successful lunar probe, bypassing the moon at 3728 miles. A year later another Soviet satellite was the first to circle the moon. On April 12, 1961, the USSR launched the first manned spacecraft, Vostok I, with Yuri Gagarin aboard. The flight made one complete circuit of the Earth at a height ranging from 112 to 187 miles, and lasted one hour and 29 minutes. Three weeks later on May 5, Alan Shepard became the first American space traveler, making a short 15-minute suborbital "hop" in a Mercury vehicle.

In the 1960s, the U.S. undertook a major program to put a man on the moon. Milestones in this project included docking maneuvers between Gemini spacecraft in December 1965; the first manned flight around the moon by

Frank Borman, James Lovell, and William Anders in December 1968; and finally, the lunar landing by Neil Armstrong and Edwin Aldrin on July 20, 1969.

Space stations have had a somewhat controversial history since they were first proposed by Tsiolkovsky in the 1920s. Stations are felt by some to be the most efficient method of launching deep space probes, because the probe vehicle will not have to overcome the force of Earth's gravity. Others feel that the cost of space stations cannot be justified in view of the marginal benefits.

Skylab, the first and only American space station, was launched on May 14, 1973 and was manned by three successive crews for a period of 171 days. The Soviets have made a greater commitment to space stations and have launched several different types, which they have manned for periods exceeding one year.

The spectacular success of the Voyager 1 and 2 probes to Jupiter, Saturn, Uranus, and Neptune indicates that the technology required to send spacecraft to the outer planets is in place.

However, for manned planetary exploration, particularly beyond Mars, it seems that some system of manned or unmanned way stations will be necessary.

This computer program, Voyage to Neptune, leaps ahead to a time when a series of manned space stations have been established on the route to the outer planets. We are assuming stations at roughly equal intervals of 400 to 500 million miles. They are at Jupiter (orbiting Callisto, the largest Jovian moon), Saturn (orbiting Titan, a moon larger than Mercury), Alpha 1 (midway between Saturn and Uranus), Uranus (on Ariel, the innermost moon), and Theta 2 (507 million miles beyond Uranus).

The voyage itself is all fiction. Your spaceship has engines which can operate using both solar energy (captured by giant parabolic collecting arrays) and nuclear fuel. Because solar radiation diminishes as you get further from the sun and you will not be able to carry enough nuclear fuel for the entire trip, you also have a multi-celled nuclear breeder reactor which takes the spent fuel from the propulsion engines along with a small amount of primary fuel and turns it all back into primary fuel.

The space stations along the way usually have a small stock of engine-repair parts (you also carry spare parts), breeder-reactor cells, and nuclear fuel, which are available to you on a barter basis.

Your objective in the game is to make the 2700-million-mile trip to Neptune in six years or less. It is possible but far from easy. Bon voyage.

THE PROGRAM

The Voyage to Neptune program consists of five main program segments, initialization and summarization segments, and six short subroutines. At 169 lines, it is a relatively short program; nevertheless, the game is challenging and entertaining.

In the initialization section (Lines 100–160), the title is displayed and variables are initialized. A seed for the random-number generator is produced in Line 1570 and reduced, if necessary, to a valid value in Line 140.

The trip-segment counter (Line 190) loops through the trip segments until the seventh destination is reached, at which time program execution branches to the End Trip segment at Line 940.

The current-conditions section (Lines 220–360) prints the current location, condition of the ship, fuel supply, and performance during the last trip segment.

The trading section (Lines 380–550) allows you to trade nuclear fuel for breeder reactor cells and vice versa. Checks are made to make sure you are not trading more than you have or that you don't trade so many cells that your breeder reactor becomes inoperative.

The engine-power section (Lines 570–620) calculates the amount of power you receive from your solar collectors. Initially solar power meets 48% of your requirements, but that power decreases by 8% during each leg of the trip. You enter the amount of nuclear fuel you want to burn, and the program checks to be sure it is not more than you have.

In the breeder-reactor usage section (Lines 640–730), you enter the number of breeder-reactor cells you wish to operate. As the breeder cells need both primary nuclear fuel (5 pounds for each cell) and spent fuel from the engines (20 pounds for each cell), these conditions must be met.

The calculations section (Lines 750–840) is the heart of the program. Efficiency of the main engines is calculated in Line 760; it is the sum of the energy from the solar collectors plus the nuclear fuel. Maximum engine output is 104%. Rate of speed in mph and total distance in millions of miles are calculated in Line 780, while Line 790 utilizes speed and distance to calculate the time (in days) of this leg of the trip.

The amount of fuel produced by the breeder-reactor cells is calculated in Line 810; not being highly efficient, they produce from 16 to 33 pounds of fuel per cell.

This new fuel is added to the total fuel. There is a 20% chance that some of the fuel in storage will decay to an unusable state; this amount, if any, is calculated in Lines 820 and 830.

There is a 10% chance of an engine malfunction (Line 770); if it occurs, you are told about the problem, and the reduced efficiency of the engines is calculated (Lines 860–920).

The end-of-trip segment (Lines 940–1160) calculates the total time it took to reach Neptune, number of operational breeder-reactor cells, and remaining amount of fuel. Appropriate messages are printed about each factor, after which you are given a chance to play again or quit.

The subroutines to print the scenario (Lines 1180–1390); read location names and distances (Lines 1410–1440); calculate and print time in years, months, and days (Lines 1460–1530); temporarily break program execution (Lines 1550–1570); and print centered lines (Lines 1680–1690) do exactly what their names describe in a straightforward way.

The subroutine to extract the first letter of an input string (Lines 1630–1660) does just that using the familiar LEFT\$ function. However, if the letter extracted is not uppercase (between A and Z), the function in Line 1660 converts the string to a numeric ASCII value, subtracts 32 (lowercase letters have ASCII values 32 higher than their corresponding uppercase letters), and converts the resulting value back to a string. While three distinct functions are performed, they can be accomplished with a single statement:

```
A$ = CHR$(ASC(A$) - 32)
```

There are several ways to make the game easier or more difficult. You can increase or decrease the propulsion efficiency of the nuclear fuel in Line 760. Dividing FUSEG by a value smaller than 40 will increase efficiency, while dividing by a value larger than 40 will decrease efficiency. In the same line, you can allow for engine speeds over 104% of normal if you wish. You can also improve the efficiency of the reactor by increasing either of the constants (16 or 18) in the first equation in Line 810:

```
FUBR = INT(16 + 18 * RND(1))
```

You can also change the barter price of fuel and breeder cells in Line 390, although doing so is a two-edged sword because sometimes you want to buy and other times to sell; best to leave it alone.

PROGRAM VARIABLES

A	Answer to input query, value
A\$	Answer to input query, string
BREED	Total breeder - reactor cells on ship
DIS(n)	Distance between segments (n = 1 to 7)
DIST	Cumulative distance traveled (million miles)
DY	Days
EF	Engine-efficiency malfunction value
EFF	Engine efficiency
FD	Fuel-decay indicator
FUBR	Fuel produced by each breeder cell
FUDCY	Fuel that decayed in storage
FUSEG	Fuel used on trip segment
FUTOT	Fuel, total available
MO	Months
RATE	Rate of speed (mph)
RN	Seed for random-number generator
PLAN\$(n)	Planet or satellite name (n = 1 to 7)
SEG	Trip-segment counter
TIME	Time of trip segment (days)
TM	Time, temporary
TOTIME	Cumulative total time of trip
UBREED	Breeder-reactor cells in use
X\$	Temporary string
YR	Years

PROGRAM LISTINGS

[Download NEPTUNE.BAS](#) (tokenized BASIC format)

```

100 CLS : KEY OFF
110 LOCATE 10, 1 : X$ = "Voyage to Neptune, 2100" : GOSUB 1680
120 LOCATE 13, 1 : X$ = "(c) by David H. Ahl, 1986" : GOSUB 1680
130 LOCATE 23, 1 : GOSUB 1550
140 WHILE RN > 32767 : RN = RN - 65535! : WEND : RANDOMIZE RN : CLS
150 GOSUB 1180 : 'Display initial scenario
160 BREED = 120 : FUTOT = 3000 : GOSUB 1420 : 'Set initial values
170 '
180 'Loop through trip segments
190 SEG = SEG + 1 : 'Trip segment counter
200 IF SEG = 7 THEN 940 : 'Reach Neptune?
210 '
220 'Print current conditions
230 PRINT : PRINT "Current conditions are as follows:"
240 PRINT " Location :" PLAN$(SEG)
250 PRINT " Distance to Neptune:" 2701-DIST "million miles."
260 IF SEG = 1 THEN 350 : 'First trip segment?
270 PRINT " Distance from Earth:" DIST "million miles."
280 PRINT "Over the last segment, your average speed was" INT(RATE) "mph,"
290 PRINT " and you covered" DIS(SEG - 1) "million miles in" TIME "days."
300 TM = .81 * DIST : PRINT "Time est for this total distance:" : GOSUB 1460
310 TM = TOTIME : PRINT "Your actual cumulative time was:" : GOSUB 1460
320 PRINT "You used" UBREED "cells which produced" FUBR "pounds of fuel each."
330 IF FUDCY = 0 THEN 350
340 PRINT FUDCY "pounds of fuel in storage decayed into an unusable state."
350 PRINT "Pounds of of nuclear fuel ready for use:" FUTOT
360 PRINT "Operational breeder-reactor cells:" BREED : PRINT
370 '
380 'Trade fuel for breeder-reactor cells
390 TRADE = INT(150 + 80 * RND(1)) : IF SEG > 1 THEN 420
400 PRINT "Before leaving, you can trade fuel for breeder-reactor cells at"
410 GOTO 430
420 PRINT "Here at " PLAN$(SEG) ", breeder cells and nuclear-fuel trade at"
430 PRINT "the rate of" TRADE "pounds of fuel per cell." : PRINT
440 IF FUTOT - TRADE < 1501 THEN PRINT "You have too little fuel to trade." : GOTO 500
450 INPUT " Would you like to procure more breeder cells (Y or N)";A$ : GOSUB 1590
460 IF A$ < ">Y" THEN 500
470 INPUT "How many cells do you want";A
480 F = FUTOT - A * TRADE : IF F > 1500 THEN FUTOT = F : BREED = BREED + A : GOTO 570
490 PRINT "That doesn't leave enough fuel to run the engines." : GOTO 470
500 INPUT "Would you like to trade some breeder cells for fuel";A$ : GOSUB 1590
510 IF A$ < ">Y" THEN 570

```

```

520 INPUT "How many cells would you like to trade";A
530 F = BREED - A : IF F > 49 THEN BREED = F : FUTOT = FUTOT + A * TRADE : GOTO 570
540 PRINT "That would leave only" F "cells. The reactor requires a minimum"

550 PRINT "of 50 cells to remain operational." : GOTO 520
560 '
570 'Engine power
580 PRINT "At this distance from the sun, your solar collectors can fulfill"
590 PRINT 56 - SEG * 8 "% of the fuel requirements of the engines. How many pounds"
600 INPUT "of nuclear fuel do you want to use on this segment";FUSEG
610 IF FUSEG <= FUTOT THEN FUTOT = FUTOT - FUSEG : GOTO 650 : 'Enough fuel?
620 PRINT "That's more fuel than you have. Now then, how many pounds" : GOTO 600
630 '
640 'Breeder reactor usage
650 INPUT "How many breeder-reactor cells do you want to operate";UBREED
660 IF UBREED > BREED THEN PRINT "You don't have that many cells." : GOTO 650
670 IF FUSEG / 20 >= UBREED THEN 700 : 'Enough spent fuel from engines?
680 PRINT "The spent fuel from your engines is only enough to operate" FUSEG / 20
690 PRINT " breeder reactor cells. Again please..." : GOTO 650
700 IF UBREED * 5 <= FUTOT THEN 730 : 'Enough new seed fuel?
710 PRINT "You have only enough fuel to seed" INT(FUTOT / 5) "breeder cells."
720 PRINT "Please adjust your number accordingly." : GOTO 650
730 FUTOT = FUTOT - 5 * UBREED
740 '
750 'Calculate the results of input data
760 EFF = 56 - SEG * 8 + FUSEG / 40 : IF EFF > 104 THEN EFF = 104 : 'Efficiency = 104% max
770 EF = RND(1) : IF EF < .1 THEN GOSUB 860 : '10% chance of engine problem
780 RATE = EFF * 513.89 : DIST = DIST + DIS(SEG) : 'Rate in mph, dist in million miles
790 TIME = INT(DIS(SEG) * 41667! / RATE) : 'Time in days
800 TOTIME = TOTIME + TIME : 'Total trip time
810 FUBR = INT(16 + 18 * RND(1)) : FUTOT = FUTOT + FUBR * UBREED : 'New fuel from breeder
820 FD = RND(1) : IF FD < .2 THEN FUDCY = INT(FD * FUTOT) : 'How much fuel decayed?
830 FUTOT = FUTOT - FUDCY : 'Decrease fuel by amount that decayed
840 GOTO 190
850 '
860 'Subroutine for engine problem
870 CLS : FOR J = 1 TO 7
880 BEEP : PRINT : X$ = "ENGINE MALFUNCTION !" : LOCATE 12, 1 : GOSUB 1680
890 FOR I = 1 TO 80 : NEXT I : CLS : FOR I = 1 TO 50 : NEXT I : NEXT J
900 PRINT "You will have to operate your engines at a" INT(300 * EF) "% reduction"
910 PRINT "in speed until you reach " PLAN$(SEG + 1) "." : PRINT
920 EFF = EFF * (1 - 3 * EF) : RETURN
930 '
940 'End of trip segment
950 PRINT : PRINT "You finally reached Neptune in"; : TM = TOTIME : GOSUB 1460
960 PRINT "Had your engines run at 100% efficiency the entire way, you would"
970 PRINT "have averaged 51, 389 mph and completed the trip in exactly 6 years."
980 IF TM > 2220 THEN 1000
990 PRINT : X$ = "Congratulations! Outstanding job!" : GOSUB 1680 : PRINT : GOTO 1070

1000 TM = TOTIME - 2190 : PRINT : PRINT "Your trip took longer than this by";
1010 GOSUB 1460 : PRINT "Your performance was"; : YR = YR + 1 : IF YR > 4 THEN YR = 4
1020 ON YR GOTO 1030, 1040, 1050, 1060
1030 PRINT "excellent (room for slight improvement)." : GOTO 1070
1040 PRINT "quite good (but could be better)." : GOTO 1070
1050 PRINT "marginal (could do much better)." : GOTO 1070
1060 PRINT "abysmal (need lots more practice)."
1070 PRINT : IF BREED < 105 THEN 1100
1080 PRINT "Fortunately you have" BREED "operational breeder-reactor cells"
1090 PRINT "for your return trip. Very good." : GOTO 1120
1100 PRINT "I guess you realize that the return trip will be extremely"
1110 PRINT "chancy with only" BREED "breeder-reactor cells operational."
1120 PRINT "With your remaining" FUTOT "pounds of fuel and" BREED "breeder"
1130 TM = 42250! / (8 + FUTOT / 40) : IF TM < 405 THEN TM = 405
1140 PRINT "cells, to get back to Theta 2 will take"; : GOSUB 1460
1150 PRINT : INPUT "Would you like to try again (Y or N)";A$
1160 GOSUB 1590 : IF A$ = "Y" THEN RUN ELSE CLS : KEY ON : PRINT "Bye!" : END
1170 '
1180 'Subroutine to print the scenario
1190 CLS : X$ = "Voyage to Neptune" : GOSUB 1680 : PRINT : PRINT
1200 PRINT " It is the Year 2100 and you are in command of the first manned"

```

```

1210 PRINT "spaceship to Neptune. Manned space stations have been established"
1220 PRINT "which orbit Callisto, Titan, and Ariel, as well as at two inter-"
1230 PRINT "mediate points between Saturn and Uranus, and Uranus and Neptune."
1240 PRINT "You must travel about 2700 million miles. At an average speed of"
1250 PRINT "over 50, 000 miles per hour, the entire trip should take about"
1260 PRINT "six years."
1270 PRINT " Your spaceship is a marvel of 21st century engineering. Since"
1280 PRINT "you may have to stop at space stations along the way, you will not"
1290 PRINT "be able to use the gravitational 'slingshot' effect of the planets."
1300 PRINT "However, your engines are highly efficient using both energy from"
1310 PRINT "the sun captured by giant parabolic arrays and nuclear fuel carried"
1320 PRINT "on board. You will not be able to carry enough fuel for the whole"
1330 PRINT "trip, so you also have a multi-celled nuclear breeder reactor"
1340 PRINT "(which takes spent fuel from your engines along with a small amount"
1350 PRINT "of primary fuel and turns it into a much greater amount of primary"
1360 PRINT "fuel)."
```

```

1370 PRINT " The space stations along the way usually have a small stock of"
1380 PRINT "engine repair parts, breeder-reactor cells, and nuclear fuel which"
1390 PRINT "are available to you on a barter basis." : GOSUB 1550 : RETURN
1400 '
1410 'Subroutine to read location names and distances
1420 FOR I = 1 TO 7 : READ PLAN$(I), DIS(I) : NEXT I : RETURN
1430 DATA "Earth", 391, "Callisto", 403, "Titan", 446, "Alpha 1", 447
1440 DATA "Ariel", 507, "Theta 2", 507, "Neptune", 0

1450 '
1460 'Subroutine to calculate and print time in years
1470 YR = INT(TM / 365) : IF YR < 1 THEN 1490
1480 IF YR = 1 THEN PRINT " 1 year"; ELSE PRINT YR "years";
1490 MO = INT((TM / 365 - YR) * 12) : IF MO < 1 THEN 1510
1500 IF MO = 1 THEN PRINT ", 1 month"; ELSE PRINT ", " MO "months";
1510 DY = INT(TM - YR * 365 - MO * 30.5) : IF DY < 1 THEN PRINT "." : RETURN
1520 IF DY = 1 THEN PRINT ", 1 day." ELSE PRINT ", " DY "days."
1530 RETURN
1540 '
1550 'Subroutine to temporarily break execution
1560 PRINT : X$ = "Press any key to continue." : GOSUB 1680
1570 WHILE LEN(INKEY$) = 0 : RN = RN + 1 : WEND : PRINT : PRINT : RETURN
1580 '
1590 'Subroutine to read yes/no answer
1600 GOSUB 1630 : IF A$ = "Y" OR A$ = "N" THEN RETURN
1610 INPUT "Don't understand your answer. Enter 'Y' or 'N' please";A$ : GOTO 1600
1620 '
1630 'Subroutine to extract the first letter of an input answer (in upper case)
1640 IF A$ = "" THEN A$ = "Y" : RETURN
1650 A$ = LEFT$(A$, 1) : IF A$ >= "A" AND A$ <= "Z" THEN RETURN
1660 A$ = CHR$(ASC(A$) - 32) : RETURN
1670 '
1680 'Subroutine to print centered lines
1690 PRINT TAB((70 - LEN(X$)) / 2) X$; : RETURN
```

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



BASIC CONVERSIONS

All of the programs in this book were written in standard Microsoft BASIC (also known as BASICA and GW BASIC) on an IBM PC-compatible computer with 512K of memory. All have been tested on a variety of PC-compatible computers.

The programs are set up for a display of 80 characters by 24 lines. All require only 72 columns of screen width, except Orient Express, which requires a full 80 columns. Thus, they cannot be easily used in a window environment unless you expand the window frame off the screen. I recommend that you run these programs directly from BASIC under MS-DOS without windows, desk accessories, or other operating-system overlays.

Some of the programs use the built-in speaker. If you have a computer with-out a speaker, don't despair; sound is not necessary to enjoy the journeys.

The memory requirements of the programs are as follows:

Program Name	Program Size in Bytes
Marco Polo	18,761
Westward Ho!	16,508
Automobile Race	20,575
Orient Express	18,321
Amelia Earhart	20,826
Tour de France	17,686
Appalachian Trail	27,392
Subway Scavenger	37,562
Hong Kong Hustle	41,506
Voyage to Neptune	7,739

All of the programs run on any IBM PC, PCjr, or PC compatible with 128K or more of memory. Remember, Microsoft BASIC itself occupies about 64K of memory; the rest of the memory space is used for the interpreted program code and data.

If you wish to convert the programs to another type of computer or another version of BASIC, be warned that this is a major undertaking. Although the programs use no unusual commands, machine language subroutines, or devious techniques, converting code from one machine to another can be tricky. If you are determined to try, here are some tips that may make your job easier:

- Memory. You should have at least 128K of memory (or at least 48K of program area after BASIC is loaded).
- Display size. You should have an 80-character by 24-line screen. Anything smaller will require changes in practically every PRINT statement.
- Enter the program one subroutine at a time. Test each one after it is entered and don't go on to the next until the current one is working. Above all, do not enter the entire program in one shot and expect it to work right away.
- Try to understand what each statement does and how it relates to other parts of the program. Refer to the list of variables and the comment statements. A thorough understanding of the program logic will make your job much easier when you reach the inevitable debugging stage.
- Depending upon the speed of your computer, timing loops may be too long or too short. Each program has a subroutine to make a short pause; these pauses should be one to two seconds long. You can experiment with different values in the FOR...NEXT loop to create an appropriate length pause.

- All the programs have a subroutine to print a centered line, specifically:

```
PRINT TAB((70 - LEN(X$)) / 2) X$ : RETURN
```

This prints whatever is in the string X\$ centered on a 70-character line. If you are converting the program to one with a smaller screen, the constant 70 should be set equal to your screen width (or a bit less for better aesthetics), and the length of X\$ cannot exceed the screen width.

Apple II Family

The main limitations in converting these programs to the Apple II family are memory size and screen size. You must have BASIC in ROM with 48K to 64K of free memory, and you must have an 80-column display.

As Applesoft BASIC does not have the IF...THEN...ELSE statement, all of the ELSE constructions must be changed. For example, in Marco Polo the following lines would have to be changed:

```
1140 IF JL > 20 THEN JL = JL - 10 ELSE JL = INT(JL / 2)
1150 GOTO 3510 : RETURN
```

Without the ELSE, these statements would become:

```
1140 IF JL > 20 THEN JL = JL - 10 : GOTO 1150
1145 JL = INT(JL / 2)
1150 GOTO 3510 : RETURN
```

Applesoft does not have INKEY\$; instead GET X\$ is used. Applesoft also does not have WHILE or WEND. All programs use INKEY\$ in a WHILE...WEND loop to halt execution temporarily. The following is one such routine:

```
3730 PRINT "Hit any key to continue."
3740 WHILE LEN(INKEY$) = 0 : WEND : RETURN
```

Here is that same routine in Applesoft BASIC.

```
3730 PRINT "Hit any key to continue."
3740 GET X$ : IF LEN(X$) = 0 THEN 3740
3745 RETURN
```

Applesoft does not have the RANDOMIZE statement to vary the starting point of the random-number generator. For RANDOMIZE, you can substitute the following statements at the beginning of each program:

```
5 INPUT "Please enter a number between 1 and 2000";A
6 FOR I = 1 TO A : RN = RND(1) : NEXT I
```

You will find a few other minor problems in converting to Applesoft, but most of them will be in the area of aesthetics rather than function. Note that while Applesoft BASIC does not have direct equivalents for some commands like CLS, LOCATE, or PRINT USING, in many cases you can achieve the same effect by using an appropriate PEEK or POKE statement.

Commodore 64, 128; Atari 8-bit Series; Tandy Color Computer

Conversion to these systems is not recommended because of their 40-column screen widths. However, if you decide to go ahead anyway, you will find the versions of BASIC on these machines substantially similar to Applesoft BASIC, so most of the above comments pertaining to the Apple II apply.

Apple Macintosh

The following comments apply only to the Microsoft BASIC interpreter for the Macintosh and not to other versions of BASIC. All of the statements and commands in Macintosh Microsoft BASIC are virtually identical to those of IBM PC Microsoft BASIC. Line numbers are not necessary on the Macintosh except when they are referred to by other statements, but you would be well advised to leave them in. Obviously, the programs do not provide for mouse support, menus, or dialog boxes.

The Macintosh display is not a constant width because characters are proportionately spaced. You may have to reformat some of the PRINT statements in your program, since you have a choice of fonts on the Macintosh. Monaco 9 point is a suggested font that helps achieve results similar to the original.

Atari ST

Atari ST BASIC is very close to Microsoft BASIC in most regards. In ST BASIC, the argument for RANDOMIZE should be zero, not RN as it is in the programs. Instead of CLS to clear the screen, Atari ST BASIC uses CLEAR 2 (clear the output window). Instead of LOCATE Y, X, Atari ST BASIC uses GOTOXY X, Y (note that the arguments are in reverse order in the two statements). In place of INKEY\$ to read a key press, the statement INP(4) should be used on the ST.

The SOUND statement of the ST is quite different from that of Microsoft BASIC. Your best bet is to leave out the sound. Or combine the information below with that in the ST BASIC Sourcebook, and work out the conversions for yourself.

MSX Computers

While MSX BASIC is virtually identical to Microsoft BASIC, these machines pose the dual problems of narrow screen width and, frequently, inadequate memory. If you have a fully expanded machine (48K or 64K) and are willing to divide most of the PRINT statements into two statements, you should have relatively few problems with BASIC itself.

Sound Routines for Microsoft BASIC

Several programs in this book utilize the speaker built into the computer to play a short melody or fanfare. The SOUND command is easy to use, but it is one of the most poorly documented commands in BASIC. The command takes the form:

`SOUND frequency, duration`

in which frequency is the desired frequency in hertz (cycles per second or cps), and duration is the time the tone is to be sounded in arbitrary units of 55 milliseconds in length. In other words, the command SOUND 100, 10 would play a 100 Hz tone for .55 seconds.

The value for frequency can range from 37 to 32767, although the upper limit of human hearing for most people is between 14,000 and 20,000 Hz. A greater limitation is the speaker built into the computer which, depending upon the computer, generally has a range of about 100 Hz to 15,000 Hz.

Duration can range from a minimum of zero to an upper limit of 32767 (30 minutes). In general, the most useful range of duration values is from 1 to 12.

If you wish to utilize the SOUND command to play a melody, it is important to know the frequency of various notes. You will have additional versatility if you know the relationship between various notes as well.

Middle A, the note most used for tuning instruments, is 440 Hz. One octave above, A is 880 Hz and one octave down, A is 220 Hz.

All notes follow this pattern, i.e., the frequency is doubled each octave.

Another important relationship is between the notes in an octave. There are 12 equally spaced notes in an octave, which are related to each other by the following formula:

$$\text{Freq}(N + 1) = \text{Freq}(N) * (2^{(1 / 12)})$$

More useful for computing is an extension of that formula for all the notes in the scale:

$$\text{Freq}(N) = \text{BaseFreq} * (2^{(N / 12)})$$

Thus the program to assign the correct frequency values to the two octaves above middle C (frequency 261.63 Hz) is as follows:

```
10 FOR I = 0 TO 24 : Freq(I) = 261.63 * (2^(I / 12)) : NEXT I
```

The following table shows the frequencies for the middle two and one-half octaves on the piano. These are the notes that are most useful for programming.

Note	Octave 1	Octave 2	Octave 3
C	262	524	
C# or D ^b	277	554	

D	294	588
D [#] or E ^b	311	622
E	165	330
F	174.5	349
F [#] or G ^b	185	370
G	196	392
G [#] or A ^b	207.5	415
A	220	440
A [#] or B ^b	233	466
B	247	494
C	262	524
		1048

TABLE 1. Note frequencies (Hertz)

Note durations are more easily related to one another than note frequencies. For example, if we assign a duration of 8 to a whole note, a half note has a value of 4, quarter note 2, eighth note 1, sixteenth note .5, and so on. A more versatile approach is to assign a whole note a duration value of DUR, a half note DUR/2, and so on. This way, you can set DUR at the beginning of the program for any tempo you want.

You can use a similar method with frequencies and assign the base frequency in Line 20 above to FRQ; then by varying FRQ, you can transpose the melody to any key you wish.

Putting all these ideas together, you will be able to add sound easily to any program. For example, here is a short program that plays the first two bars of "Give My Regards to Broadway."

```

10 DIM FREQ(25)
20 FRQ = 262 : DUR = 12
30 FOR I = 0 TO 24 : '2 OCTAVES
40 FREQ(I) = FRQ * (2^(I / 12)) : 'FILL NOTES WITH CORRECT FREQUENCIES
50 NEXT I
60 FOR I = 1 TO 15
70 READ NOTE, LN : 'READ NOTES AND LENGTHS FROM DATA
80 SOUND FREQ(NOTE), DUR/LN : 'PLAY NOTES
90 NEXT I
100 DATA 0, 3, 2, 2, 4, 3, 5, 2, 7, 2, 5, 1, 4, 1
110 DATA 4, 4, 2, 4, 2, 4, 2, 4, 2, 4, 4, 2, 4, 2, 0, 1

```

If It Doesn't Work

When a program blows up, your first reaction may be to kick your computer and yell, "There ought to be a law against people who sell books full of programs that don't work!" However, after many years of experience, we have found that almost all the problems people have with published programs result from simple typing mistakes or lack of familiarity with the peculiarities of their machines. So if a program doesn't work, we strongly suggest that you have someone else check your program character-by-character or line-by-line against the one in this book. The reason for having someone else do it is that if you have typed something wrong because you read it incorrectly, you will tend to continue to see it incorrectly.

If you are having a problem with one of the programs, reason it through. For example, in Westward Ho!, if you always run out of food on the first trip segment, that would suggest that the starting quantity is too small. When the supplies-buying routine is executed, perhaps instead of multiplying the quantity per dollar by the dollars spent, the quantities have been added by mistake. Or maybe there is a typo in the variable name. Every program has a list of variables, and practically every statement in the listing is commented; thus, you should be able to pinpoint an error once you think it through.

If, after following these procedures, you identify what you are convinced is a genuine bug, please do the following:

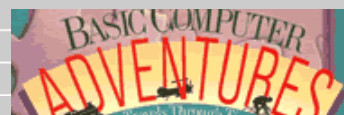
- Describe the problem completely in writing.
- If you have a line printer, list the entire program, not just the statement you think is in error. Also, using the screen print feature, print one or more output screens that illustrate the problem.
- Send the above items with a self-addressed stamped envelope to: Basic Adventures Problem, David Ahl, 12 Indian Head Road, Morristown, NJ 07960.

Please note that I am absolutely unable to handle telephone inquiries. Please understand that there are well over one million copies of my books in circulation and if just one-tenth of one percent of the readers called me for a half-hour debugging session...well, you get my point. ~~I will, however, respond to all written inquiries that adhere to the guidelines given above.~~

Bugs aside, I'm sure you will devise many ways to extend and improve the programs in this book. I would be pleased to hear from readers—in writing—with comments about this book and, most especially, with suggestions for programs you would like to see in my next book.

Until then, have fun trekking through this volume.

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



About the Author

David Ahl is the well-known author of more than 20 books covering the educational and recreational applications of computers, including the best-selling *BASIC Computer Games* and *More BASIC Computer Games*. In 1974, he founded *Creative Computing* magazine, the first consumer magazine devoted to personal computers. David has also written for such publications as *National Geographic World*, *Travel Weekly*, *Omni*, *Science Digest*, *The New York Times*, and *The Christian Science Monitor*.

The manuscript for this book was prepared and submitted to Microsoft Press in electronic form. Text files were processed and formatted using Microsoft Word.

Cover and interior text design by Becker Design Associates

Cover illustration by Scott McDougall

Principal typographer: Christopher Banks

Principal production artist: Becky Johnson

The screen displays were created on the IBM Personal Computer and printed on the Hewlett Packard LaserJet.

Text composition by Microsoft Press in Baskerville and display in Univers 49, using the CCI-400 composition system and the Mergenthaler Linotron 202 digital phototypesetter.

[Return to Table of Contents](#) | [Previous Section](#) | [Next Section](#)



Download Games from BASIC Computer Adventures

The games are stored on the distribution disk in tokenized BASIC format. They run under standard PC BASIC or Microsoft GW BASIC. The READ_ME file has instructions for running these programs - they should work fine on an old PC with BASIC or an emulator. These programs are now freeware: you may ignore the shareware notices.

- All games in ZIP file [Download](#)
- or download individual programs:
 - Marco Polo [Download](#)
 - Westward Ho! [Download](#)
 - The Longest Automobile Race [Download](#)
 - The Orient Express [Download](#)
 - Amelia Earhart: Around the World Flight [Download](#)
 - Tour de France [Download](#)
 - Appalachian Trail [Download](#)
 - Subway Scavenger [Download](#)
 - Hong Kong Hustle [Download](#)
 - Voyage to Neptune [Download](#)
 - Menu program [Download](#)
 - Readme file [Download](#)

[Return to Table of Contents](#) | [Previous Section](#)