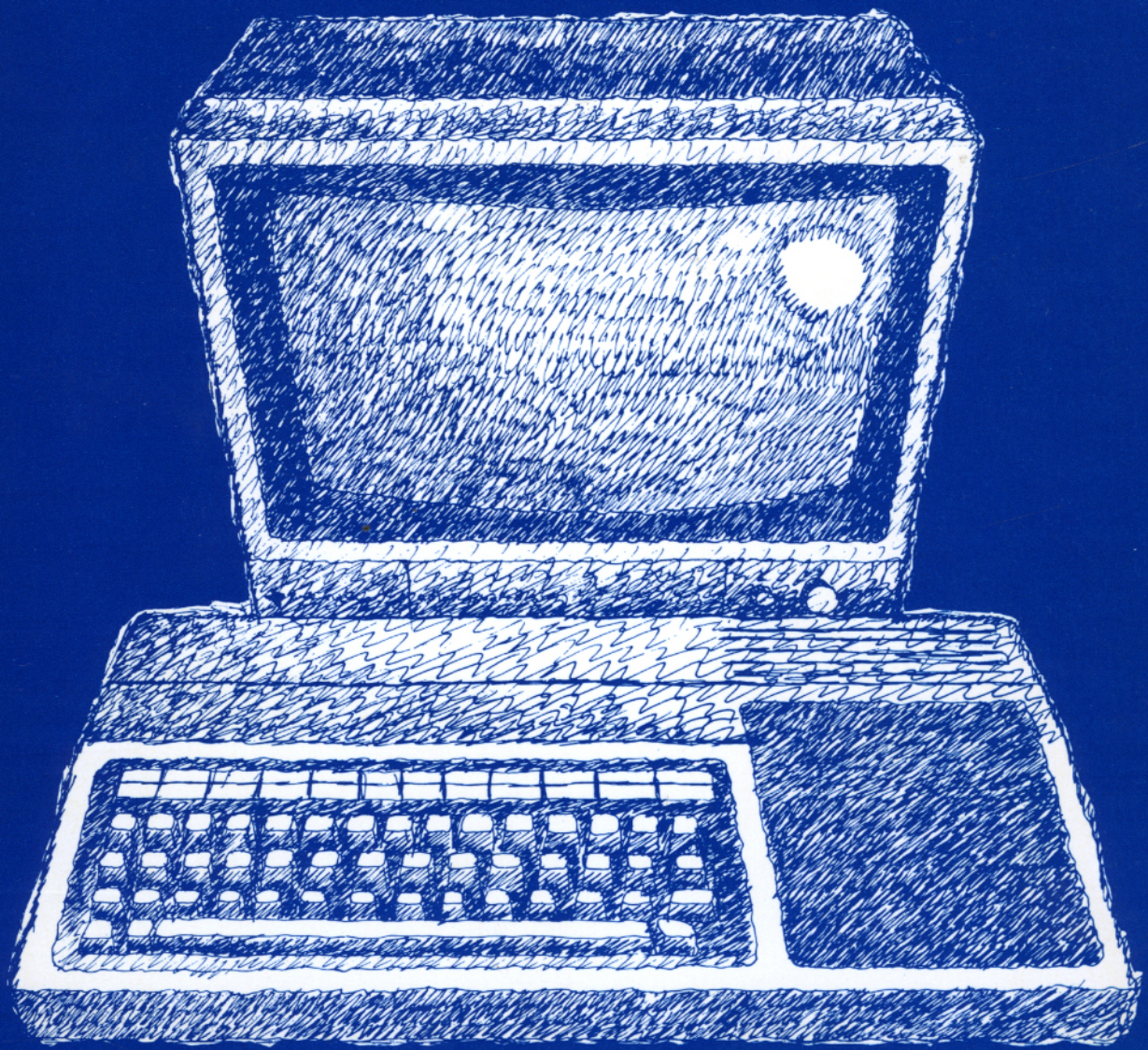
Programs for the TI Home Computer

by Steve Davis



Programs for the TI Home Computer

by Steve Davis

FIRST EDITION

Steve Davis Publishing
P.O. Box 190831
Dallas, Texas 75219

ISBN #0-911061-00-2
Library of Congress Catalog Card Number 82-90783

TABLE OF CONTENTS

Table of Contents

PREFACE

The Texas Instruments Home Computer is one of the fastest selling personal computers on the market today, and for good reason. It is a remarkable device. Though it costs less than most color television sets and is relatively compact and lightweight, it works magic that was not possible a little over 11 years ago when Texas Instruments introduced the first microprocessor to the world. Colorful graphics, human-like speech, and music are just the icing on a very useful cake. Hundreds of thousands of people are using the TI 99/4 and 99/4A computers at home, at school or at work. Yet, despite the popularity of the unit, and despite the recent proliferation of books on computers, there has not been a book of BASIC programs published especially for users of the TI Home Computer. Until now, that is.

This book is exactly what the title implies: a collection of programs written in TI BASIC and TI Extended BASIC. It is not a book *about* TI BASIC but rather a book *of* TI BASIC. However, it has been said by others (and bears repeating) that one of the best ways to learn a computer language is to use it. After all, that is how we learn our native language when we are young; we observe how the language is used by others until we become comfortable enough with it to use it ourselves, sometimes adding to it our own individual styles of expression. Thus, the beginner can learn a lot about programming by typing in programs written by others and observing what BASIC statements make the program work.

Further, by entering the program yourself, you are free to modify it in any way that suits your needs. A majority of the listings in this book were written to run in TI BASIC with only the 16K console and a cassette recorder so as to accommodate beginning users. However, most of the programs can be enhanced to fit your particular system configuration. Some programs operate much more efficiently in Extended BASIC, so they are listed in that way. The version of each program, and in most cases, the peripheral equipment required, if any, is printed in the listings.

I have tried to include programs that use the TI computer to its best advantage by utilizing color, graphics, music, sound or speech in many of the programs. Because of the differences in the BASIC language from one brand of computer to another, most books of programs are sadly lacking in these little embellishments that make a program more fun to use.

A few acknowledgements are in order. I would like to thank Don Cook, Mike Wilcox, David Migicovsky, Brian Madigan and John Clulow for their program contributions, Nick Georgopoulos for the typesetting, and David Parnell for having answers to my questions about the mysteries inside the 99/4A.

Now, just two closing REMarks. First, this book is not intended to be the last word in TI BASIC programs. On the contrary. It is my hope that the publication of this

humble effort will start a trend towards filling the gap of printed information needed by users of the TI Home Computer. Finally, these programs are not perfect! Perhaps, while typing in a listing, you will come across a statement or a routine that makes you say to yourself, "I would have done this another way." If so, you should share your ideas with other users, for they in turn may have ideas to give you. The constant exchange of ideas and information between computer users is, to me, one of the most stimulating aspects of owning a home computer. But perhaps, you may find something in this book that will turn on a little light bulb somewhere in your mind (your personal RAM), and you may say to yourself, "Oh, I see!" And, perhaps it will stimulate you to write your own program, perhaps a new application for your computer or a new way to do something more effectively. If there is but one line in one program in this volume that inspires you to create your own ideas or to appreciate your computer more, then this book will have fulfilled its purpose.

Happy Computing,

Steve Davis

INTRODUCTION

Before typing in any TI BASIC program, you should read through the *Users Reference Guide* (and any Addenda) that accompanies your TI Home Computer. Those with TI Extended BASIC should also become familiar with the special statements used in that language by reading the TI Extended BASIC manual (and Addenda). Key in the examples used in those manuals so you will understand how each statement works. Practice using the special function keys and edit functions.

Those with the TI 99/4 Computer may only use upper-case letters. In the very few places where lower-case is used in programs contained in this book, upper-case may be substituted in most instances. However, if lower-case characters are being used to display redefined graphics characters, you may use the CHR$ function. For example, PRINT CHR$(97) is the same as PRINT"a". If using the 99/4A, remember to keep the alpha-lock key depressed when entering statements unless the program specifically calls for lower-case.

If you have the older version of Extended BASIC, you will not be able to use lower case unless you define those characters. There is a program in this book which defines the lower case characters for any TI BASIC or Extended BASIC program.

Enter program lines *exactly* as they are listed. Notice that the number zero ("0") is printed with a slash through it in the listings to differentiate it from the letter "O". Include the proper number of spaces between words. Some print statements and Data statements have been especially formatted for printing on the 28-column text screen. There may be a few unusually long program lines that will require use of the Edit and Insert functions. Normally TI BASIC allows a program line to be up to 4 screen lines long, and TI Extended BASIC allows 5-line statements. However, if you edit the line (by typing the line number & pressing function up or down arrow on the 99/4A), you may sometimes insert characters to increase the length of the statement.

Special attention should be paid to accuracy when entering Data statements. The comma is used to separate pieces of data, so if you put in an extra one or leave one out, the correct data will not be read. When putting your own data statements into programs, such as some of the record-keeping programs in this book, be sure to list all data in the sequence specified, including any special symbols that serve to mark the beginning or end of Data statements.

When keying in a program in Extended BASIC, notice that a single colon represents a print separator (carriage return), while a double-colon (":") indicates a statement separator in multiple-statement lines. Place a space between colons if they are to be used as print separators. Check your typing for accuracy as you enter each line. It is easier to catch errors that would cause your program to "crash" at that point than later. After entering several lines, SAVE the program on

tape or disk in case a mistake or malfunction erases the program in memory. When you have finished entering the program and have SAVEd it on tape or disk, make a back-up copy before running it. If the program will not run properly, usually you will receive an error message specifying the line number. LIST that line and check your typing very carefully. Also, check line number references in GOTO, GOSUB and IF-THEN statements.

These programs may seem rather "un-REMarked" in the sense that the number of REMark statements has been kept to a minimum to make typing easier and to save memory. It is a good idea to key in all the REMarks that appear because they are important in documenting the program. The "lead" REMarks are especially important because they identify the program and its source, in case you ever need to refer back to the listing or documentation. If a program is very long, those using disk may have to execute a CALL FILES(1) statement before loading it. Any program can be altered to suit your taste, but this should only be done after you have typed in the program as written and have checked to be sure you have made no errors.

ALLIGATOR ALLEY

Can you find your way out of the murky swamp without getting eaten by an alligator? This game is so simple that it can even be played by small children. Every time you make a move you have a 1 in 20 chance of being eaten. Try to move your marker to any edge (rows 1 or 24 or columns 1 or 32 on your monitor) without stepping on any alligators. Remember that alligators are camouflaged in the swamp, so with every step, you risk your life!

To move your marker, just press E, X, D, or S, for North, South, East, or West respectively. (These are the arrow keys on the TI-99/4A, but you do not need to press the Function key).

The 24 by 32 array, A(R,C), keeps track of the 768 character positions on the screen. If A(R,C)=X, there is an alligator waiting there. If you wish to alter your odds for survival, simply change the number being multiplied by RND in lines 190 and 370. (The number must be the same in both lines). For example, making X and A(R,C) equal to INT(RND*30)+1 would improve your chances of survival by 50%, while INT(RND*10)+1 would double your risk.

Though the program is short, the array uses more memory than the program listing itself. This is due to the fact that the computer uses up to 9 bytes to keep track of the integer values of each location, unlike strings, which consume only one byte each plus a byte for each character. If you wish to play another round, the random number X is changed to relocate the hungry amphibians. Thus, once the array has been loaded at the start of the game, you do not need to wait for it to be initialized again, and each game will be different.

Here's to your survival!

```
100 REM *ALLIGATOR ALLEY* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM A(24,32)
140 RANDOMIZE
150 CALL CLEAR
160 PRINT "I'M HIDING THE ALLIGATORS":"STAND BY"
170 FOR R=1 TO 24
180 FOR C=1 TO 32
190 A(R,C)=INT(RND*20)+1
200 NEXT C
210 NEXT R
220 A(12,16)=0
230 CALL CHAR(96,"FFFFFFFFFFFFFFFF")
240 CALL COLOR(9,16,1)
250 PRINT :"YOU ARE IN THE MIDDLE OF A  SWAMP.  YOU MUST REACH THE  EDGE WITHOUT
 GETTING EATEN  BY AN ALLIGATOR."
260 PRINT :"MOVE YOUR MARKER TO ONE SIDEBY PRESSING THE ARROW KEYS: S=WEST, D=EA
ST, X=SOUTH,     E=NORTH"
270 PRINT :"ALLIGATORS ARE GREEN SO THEYARE CAMOUFLAGED IN THE SWAMP": :"HOPE YO
U DON'T GET EATEN!"
```

```
280 PRINT :"PRESS ANY KEY TO START"
290 CALL KEY(0,KEY,STATUS)
300 IF STATUS=0 THEN 290
310 CALL CLEAR
320 CALL SCREEN(3)
330 CALL HCHAR(12,16,96)
340 R=12
350 C=16
360 M=0
370 X=INT(RND*20)+1
380 CALL KEY(0,KEY,STATUS)
390 IF STATUS=0 THEN 380
400 IF (KEY()69)*(KEY()101)THEN 440
410 IF R=1 THEN 610
420 R=R-1
430 GOTO 550
440 IF (KEY()88)*(KEY()120)THEN 480
450 IF R=24 THEN 610
460 R=R+1
470 GOTO 550
480 IF (KEY()83)*(KEY()115)THEN 520
490 IF C=1 THEN 610
500 C=C-1
510 GOTO 550
520 IF (KEY()68)*(KEY()100)THEN 380
530 IF C=32 THEN 610
540 C=C+1
550 CALL HCHAR(R,C,96)
560 M=M+1
570 IF A(R,C)=X THEN 660
580 FOR D=1 TO 20
590 NEXT D
600 GOTO 380
610 CALL SOUND(1000,262,0,330,0,392,0)
620 PRINT "WHEW! YOU MADE IT IN";M+1;"MOVES!";"WANT TO TRY AGAIN? (Y/N)"
630 INPUT Y$
640 IF (Y$="Y")+(Y$="y")THEN 310
650 STOP
660 ROW=R
670 COL=C
680 IF ROW(24 THEN 700
690 ROW=23
700 IF COL)2 THEN 720
710 COL=3
720 IF C(29 THEN 740
730 COL=28
740 CALL CLEAR
750 CALL SOUND(900,110,2,-7,6)
760 CALL SCREEN(16)
770 CALL COLOR(10,13,1)
780 CALL CHAR(104,"000000000000703F")
790 CALL CHAR(105,"0F03071F38")
800 CALL CHAR(106,"00000000000000E3")
810 CALL CHAR(107,"7FFFFFFF0F070307")
820 CALL CHAR(108,"00007C3F070100E")
830 CALL CHAR(109,"FCFFFFFFFFFF8081")
840 CALL CHAR(110,"000000C0F0FC7E3E")
850 CALL CHAR(111,"3FFFFEFEFCF8E0E")
860 CALL HCHAR(ROW,COL,104)
870 CALL HCHAR(ROW+1,COL,105)
880 CALL HCHAR(ROW,COL+1,106)
890 CALL HCHAR(ROW+1,COL+1,107)
900 CALL HCHAR(ROW,COL+2,108)
910 CALL HCHAR(ROW+1,COL+2,109)
920 CALL HCHAR(ROW,COL+3,110)
```

```
930 CALL HCHAR(ROW+1,COL+3,111)
940 FOR I=1 TO 5
950 CALL CHAR(104,"0000000000000001")
960 CALL CHAR(105,"3F033F7F")
970 FOR D=1 TO 10
980 NEXT D
990 CALL CHAR(104,"000000000000703F")
1000 CALL CHAR(105,"0F03071F38")
1010 FOR D=1 TO 10
1020 NEXT D
1030 NEXT I
1040 CALL CLEAR
1050 PRINT "OH NO! YOU WERE EATEN!": :"WANT TO TRY AGAIN? (Y/N)"
1060 GOTO 630
```

CHAOS

For fans of jumbled word puzzles, this TI BASIC game has fifty words which are chosen and scrambled randomly. Of course, you can substitute your own word list by changing the Data statements. To use as an educational game, for example, you could use a list of words pertaining to a particular subject. You might want to ask a friend to type in the data so you won't know which words are being scrambled.

```
100 REM *CHAOS* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM A$(50)
140 CALL CLEAR
150 RANDOMIZE
160 PRINT TAB(8);"LET'S  PLAY": : :TAB(9);"H O S A C": :
170 FOR X=1 TO 20
180 CALL SOUND(50,RND*1650+110,3)
190 NEXT X
200 PRINT TAB(8);"^ ^ ^ ^ ^ ^";TAB(6);"^ ^ ^ ^ ^ ^ ^ ^";TAB(4);"^ ^ ^ ^ ^ ^ ^ ^
^ ^";
210 PRINT :TAB(6);") C H A O S ! (": :TAB(6);"_ _ _ _ _ _ _ _ _": : : :
220 GOSUB 710
230 PRINT "THE COMPUTER WILL GIVE YOU  A MIXED-UP WORD.":"SEE HOW MANY WORDS YOU
 CAN  UNSCRAMBLE."
240 PRINT :"INITIALIZING--STAND BY"
250 FOR I=1 TO 50
260 READ A$(I)
270 NEXT I
280 FOR I=1 TO 50
290 R=INT(RND*50)+1
300 X$=A$(I)
310 A$(I)=A$(R)
320 A$(R)=X$
330 NEXT I
340 CALL CLEAR
350 IF Q>=50 THEN 620
360 Q=Q+1
370 PRINT "SCRAMBLING WORD #";Q ::
380 C$=""
390 FOR I=1 TO LEN(A$(Q))
400 B$(I)=SEG$(A$(Q),I,1)
410 NEXT I
```

```
420 FOR I=1 TO LEN(A$(Q))
430 R=INT(RND*5)+1
440 X$=B$(I)
450 B$(I)=B$(R)
460 B$(R)=X$
470 NEXT I
480 FOR I=1 TO LEN(A$(Q))
490 C$=C$&B$(I)
500 NEXT I
510 PRINT "UNSCRAMBLE THIS WORD:": : :
520 PRINT C$
530 INPUT G$
540 IF G$=A$(Q)THEN 570
550 PRINT "SORRY, THE WORD IS":A$(Q): :
560 GOTO 600
570 PRINT : :"CONGRATULATIONS, YOU WIN!": : :
580 V=V+1
590 GOSUB 650
600 INPUT "ANOTHER WORD? (Y/N)":Y$
610 IF Y$="Y" THEN 340
620 PRINT :"OUT OF ";Q;" WORDS":"YOU GOT ";V;" CORRECT."
630 PRINT "THAT'S A SCORE OF ";INT(V/Q*100+.5);"%"
640 STOP
650 CALL SOUND(100,440,2)
660 CALL SOUND(100,550,2)
670 CALL SOUND(100,440,2)
680 CALL SOUND(100,660,2)
690 CALL SOUND(100,880,2)
700 RETURN
710 PRINT : : : :"      PRESS ANY KEY"
720 CALL KEY(0,KEY,STATUS)
730 IF STATUS=0 THEN 720
740 CALL CLEAR
750 RETURN
760 DATA PREVAIL,ACADEMIC,COMPANY,DIAPER,PROXIMITY,SQUEEZE,BAFFLE,OCCUPANT,LEVIT
Y,SWELLING
770 DATA PARASITE,MEMORY,LOCATION,LIQUIDS,REGISTER,ANIMAL,FLEXIBLE,NATIONAL,MARG
ARINE,OPINION
780 DATA SPIRITUAL,RIDDLE,UNAWARE,QUICKLY,WICKED,PROGRAMMER,RECORDED,PEACHES,BEL
LY,PROFUSE
790 DATA PHOTOGRAPH,NEPHEW,DINOSAUR,ESKIMO,DIPLOMA,SERIOUS,SIGNALS,SILHOUETTE,RO
DEO,ROUTINE
800 DATA PROMINENT,NEGLECT,SKILLFUL,HIGHWAY,FABRICATE,ARRANGE,UNSCRAMBLE,CRITICA
L,SAFETY,NOBODY
```

CHARADES

Remember the "good old days" when you had friends over for a party and you all ended up playing games? These days, you probably have your guests lined up at your computer to take turns at Invaders or some other such one-man game. At last, here is a chance to use your computer to help you entertain the whole group with the old parlor game Charades. Charades has always been popular as a party game because it allows many players to participate, it is lively and fast-moving, and it generally lends itself to a good time. It even allows your slightly tipsy guests to make absolute fools of themselves in a perfectly constructive context.

As well as being a wild and crazy adult game, Charades can be great entertainment for the youngsters as well. By altering the Data statements in the program, a custom list of words or phrases on any number of subjects can be substituted, making the program quite versatile as an educational tool.

In the original game, you needed someone to keep time and score and you had to take time to think of phrases, write them on slips of paper, and draw them out of a hat. Now, the computer can take care of all those chores for you. For those who have never played, and for those who are just a little rusty, here is a brief rundown of the rules:

Although as few as 2 can play, it is suggested that an even number of players of 6 or more participate because the group will be divided into 2 teams. Players from each team take turns pantomiming phrases to be guessed by the other members of their team. Phrases may be broken into words or syllables, but the player may not talk, write or form words with his lips while he is pantomiming. He has 2 minutes to convey the phrase to his teammates, and the time he uses determines his score.

A player should begin by pantomiming the category of the phrase. Categories used in this program include Movies, Books, People, Songs, Quotes and Cliches. Some of the traditional signals for these categories in Charades are:

Movie — Hold one hand in front of your face and turn the other one in a circle, as if cranking an old-time movie camera.
Book — Put hands together, as if praying, then open them like a book.
Song — Hold arms out and open mouth, as if singing.
Person — Pat yourself on the head.
Quote — Hold arms out with 2 fingers out on each, as if putting quotes around something.

Holding up a certain number of fingers indicates the number of words in the phrase, which word you are acting, or the number of syllables in a word. Pinching your ear means that the word you are acting "sounds like" the one in your phrase. If you are creative, you will be good at this game.

This program displays the phrase for each player to study before he pantomimes it, so situate your TV screen so that your teammates cannot see it. However, turn up the volume because the program provides an audible "times up" tone (just like the one in those TV game shows that so rudely informs the contestant that, indeed, she did *not* win the washer and dryer).

The program listed here is written in TI BASIC. It takes advantage of several special routines that the TI offers, including sound capabilities that not only provide audible prompts but also make timing loops as accurate as possible.

If you get hooked on this game, you may want to substitute your own phrases for variety. Adding words that are of interest to your group (*i.e.* computers), foreign words, or even X-rated terms, presents all kinds of possibilities. So, why not let your micro liven up your next party with this new slant on an old game?

Line 120 sets up arrays for 125 phrases (M$) and a counter to check for duplications (Z). Line 120 assures a different set of random numbers for each game, and the counter "Q" keeps track of how many phrases have been played. The subroutine at 1460 (referred to in line 160) plays the song "Charade." At 280, the GOSUB 1190 waits for the player to press a key before clearing the screen and moving on to the next routine. The loop at 530 reads the 125 phrases into the array (M$), and the loops at 320 and 340 assure that each team has alternating turns and that 5 rounds make a game. To shorten or lengthen the game, change the number of rounds (5) in 320.

The routine beginning at 950 first indicates whose turn it is, then it generates a random number (X) which, when used as a subscript to M$, will select which phrases will be played next. GOSUB 1380 checks to see if the phrase selected has been used so as to avoid duplication of phrases during the game. In other words, once it is "drawn out of the hat" it is discarded and can't be used again. (It is unlikely that you will play long enough to use all the phrases, but after about 100 have been used, you will naturally notice that it takes a bit longer for the computer to select an unused phrase). At 1010 the program determines and prints the category of the phrase (there are 5 groups of 25 phrases); the phrase itself is printed at 1140. Again, the routine at 1190 is used to wait for a signal from the player to clear the screen and start the clock.

The routine at 1280 is the clock, which counts down the time (T) and thus the score. The first CALL SOUND statement in the loop (line 1300) plays an inaudible tone (40,000 hz) at -30 db for 750 milliseconds (¾ of a second), then line 1310 gives the clock a "tick" by sounding a short (20 millisecond) 220 hz tone at -10 db. CALL SOUND was used as a timing device because it can be more accurately adjusted than delay loops; however, a FOR-NEXT loop of, say, 1 to 250 might be used instead at line 1300. If you hold down a key when your phrase has been guessed, the clock will stop, thanks to lines 1320 and 1330, and the last number displayed

(T) becomes your score for that round. Each loop takes a total of 1 second. To give players more or less than the 2 minutes allowed here, change the number (120) in line 1280. Lines 1350 and 1360 provide a loud "times up" tone and reprint the phrase.

After 5 rounds, a C-major 3-note fanfare (at lines 1240-1260) announces the end of that game. The score of the winning team is displayed. If you have played 12 games (and by that time it should be well past your bedtime), lines 490 and 510 end the game before you run out of data.

The whole idea of Charades, of course, is to convey an idea without speaking. But if you have the Texas Instruments Speech Synthesizer unit and either the Speech Editor module or the TI Extended BASIC module, then it is okay to let your computer do the talking. If you wish, try adding these lines to the program:

```
185 CALL SAY("DO YOU WANT INSTRUCTIONS")
275 CALL SAY(P$)
335 CALL SAY("THIS IS ROUND NUMBER")
336 CALL SAY(STR$(ROUND))
425 CALL SAY("NUMBER 2 #YOU WIN#")
465 CALL SAY("NUMBER 1 #YOU WIN#")
475 CALL SAY("DO YOU WANT TO PLAY AGAIN")
515 CALL SAY("GAMES OVER. GOODBYE")
965 CALL SAY("NUMBER")
966 CALL SAY(STR$(TEAM))
967 CALL SAY("IT IS YOUR TURN")
968 CALL SAY(P$)
969 CALL SAY("TO SEE YOUR WORDS")
1145 CALL SAY(P$)
1146 CALL SAY("TO START")
1165 CALL SAY(P$)
1166 CALL SAY("TO STOP. GO NOW")
1355 CALL SAY("STOP YOUR TIME IS UP")
```

(This program originally appeared in COMPUTE magazine.)

```
100 REM *CHARADES* TI BASIC, FROM PROGRAMS FOR THE TI HOME COMPUTER
110 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
120 DIM M$(125),Z(125)
130 RANDOMIZE
140 CALL CLEAR
150 PRINT TAB(8);"* CHARADES  *":::::::::::::
160 GOSUB 1460
170 P$="      PRESS ANY KEY"
180 Q=1
190 INPUT "WANT INSTRUCTIONS? (Y/N)":Y$
200 IF Y$<>"Y" THEN 290
210 PRINT :"DIVIDE GROUP INTO 2 TEAMS.":"YOU WILL PLAY 5 ROUNDS EACH.":"WHEN INS
TRUCTED, 1 PLAYER"
```

```
220 PRINT "FROM TEAM# DISPLAYED SHALL":"PRESS A KEY TO REVEAL HIS":"PHRASE. HE S
HOULD STUDY IT"
230 PRINT "BEFORE PRESSING A KEY TO":"START CLOCK. HE HAS 120 SEC.":"TO PANTOMIM
E THE CATEGORY"
240 PRINT "& PHRASE TO HIS TEAM.":"HE MAY NOT TALK OR WRITE.":"WHEN THE PHRASE I
S GUESSED,"
250 PRINT "HOLD DOWN A KEY UNTIL CLOCK":"STOPS. A TONE WILL SOUND":"WHEN TIME IS
 UP."
260 PRINT "THE LESS TIME YOU USE,":"THE HIGHER YOUR SCORE."
270 PRINT "CATEGORIES INCLUDE MOVIES,":"SONGS,BOOKS,PEOPLE,AND":"QUOTES & CLICHE
S."
280 GOSUB 1190
290 GOSUB 530
300 SCOR(1)=0
310 SCOR(2)=0
320 FOR ROUND=1 TO 5
330 PRINT "ROUND #";ROUND: : :
340 FOR TEAM=1 TO 2
350 GOSUB 950
360 SCOR(TEAM)=SCOR(TEAM)+T
370 NEXT TEAM
380 NEXT ROUND
390 GOSUB 1240
400 IF SCOR(1))SCOR(2)THEN 460
410 IF SCOR(1)=SCOR(2)THEN 440
420 PRINT : :"CONGRATULATIONS ,TEAM #2!": :"YOU WIN WITH A SCORE OF":SCOR(2)
430 GOTO 470
440 PRINT : :"IT'S A TIE! THAT DOESN'T    HAPPEN OFTEN!"
450 GOTO 470
460 PRINT : :"CONGRATULATIONS, TEAM #1!": :"YOU WIN WITH A SCORE OF":SCOR(1)
470 PRINT : :"WANT TO PLAY AGAIN? (Y/N)"
480 INPUT Y$
490 IF Q)=120 THEN 510
500 IF Y$="Y" THEN 300
510 PRINT "GAME OVER. OUT OF DATA":"TYPE RUN TO START AGAIN"
520 END
530 PRINT "INITIALIZING DATA,STAND BY"
540 FOR I=1 TO 125
550 READ M$(I)
560 NEXT I
570 CALL CLEAR
580 RETURN
590 REM *MOVIES*
600 DATA A MAN AND A WOMAN,MAN WITH THE GOLDEN ARM,SOME LIKE IT HOT,MARY POPPINS
610 DATA WHITE CHRISTMAS,MUTINY ON THE BOUNTY,ON THE WATERFRONT,YOUNG FRANKENSTE
IN
620 DATA AGONY AND THE ECSTASY,THE WIZARD OF OZ,YOU ONLY LIVE TWICE,THE LITTLE F
OXES
630 DATA DIAL M FOR MURDER,NORTH BY NORTHWEST,PSYCHO,LADY SINGS THE BLUES
640 DATA MEET ME IN ST.LOUIS,THE GREAT ZIEGFELD,LAURA,THE EMPIRE STRIKES BACK
650 DATA WHERE THE BOYS ARE,DOCTOR ZHIVAGO,DOCTOR STRANGELOVE,2001 A SPACE ODYSS
EY,THE TURNING POINT
660 REM *BOOKS*
670 DATA VALLEY OF THE DOLLS,THE CARPETBAGGERS,GONE WITH THE WIND,EVERYTHING YOU
 WANTED TO    KNOW ABOUT SEX
680 DATA CATCHER IN THE RYE,THE BIBLE,MAGNIFICENT OBSESSION,OLIVER TWIST
690 DATA WOMEN IN LOVE,JANE EYRE,REBECCA,ALICE IN WONDERLAND
700 DATA THE HOBBIT,FUTURE SHOCK,GOODBYE MR.CHIPS,MOBY DICK
710 DATA HUCKLEBERRY FINN,WAR AND PEACE,LITTLE WOMEN,GULIVERS TRAVELS
720 DATA BRAVE NEW WORLD,THE SCARLET LETTER,TALE OF TWO CITIES,GIANT,LOLITA
730 REM *PEOPLE*
740 DATA MARILYN MONROE,MARIE ANTOINETTE,GROUCHO MARX,JOHN KENNEDY
750 DATA MARTIN LUTHER KING,SOPHIA LOREN,WALTER CRONKITE,SEAN CONNERY
760 DATA ELEANOR ROOSEVELT,JUDY GARLAND,EDGAR HOOVER,COLUMBUS
770 DATA GREER GARSON,RONALD REAGAN,LADY BIRD JOHNSON,NELSON EDDY
```

```
780 DATA JOHNNY CARSON,GEORGE WALLACE,CYD CHARISSE,GRETA GARBO
790 DATA DOLLY PARTON,JOAN CRAWFORD,BETTE DAVIS,PAT NIXON,GEORGE GERSHWIN
800 REM  *QUOTES&CLICHES*
810 DATA A STITCH IN TIME SAVES NINE,DONT LOOK A GIFT HORSE IN   THE MOUTH,CLEAN
 AS A WHISTLE,NEVER SAY DIE
820 DATA REMEMBER THE ALAMO,IGNORANCE IS BLISS,HASTE MAKES WASTE,CONTENTED AS A
COW
830 DATA ALL THAT GLITTERS IS NOT    GOLD,PURR LIKE A KITTEN,I SHALL RETURN,SHAR
P AS A TACK
840 DATA TO BE OR NOT TO BE,I'LL THINK ABOUT THAT       TOMORROW,I WANT TO BE AL
ONE,THE BUCK STOPS HERE
850 DATA WE HAVE NOTHING TO FEAR BUT FEAR ITSELF,THAT'S ALL FOLKS,WHAT'S UP DOC?
,THERE'S NO PLACE LIKE HOME
860 DATA DONT COUNT YOUR CHICKENS TIL THEY HATCH,PARTING IS SUCH SWEET SORROW,HO
LD YOUR HORSES
870 DATA IT'S ALWAYS DARKEST BEFORE  THE DAWN,HINDSIGHT IS 20/20 VISION
880 REM *SONGS*
890 DATA SANTA CLAUS IS COMING TO    TOWN,STARDUST,MY FUNNY VALENTINE,FEELINGS
900 DATA MIDNIGHT BLUE,PEOPLE,CAMP TOWN RACES,SOME ENCHANTED EVENING
910 DATA DO RE MI,I WANNA HOLD YOUR HAND,YESTERDAY,DOWNTOWN
920 DATA HOUSE OF THE RISING SUN,MY COUNTRY TIS OF THEE,THE LADY IS A TRAMP,THE
MAN I LOVE
930 DATA ST.LOUIS BLUES,AMERICAN PIE,STORMY WEATHER,OVER THE RAINBOW
940 DATA YOU'VE GOT A FRIEND,MOON RIVER,I GOT PLENTY OF NOTHIN,TRY TO REMEMBER,Y
OU'LL NEVER KNOW
950 REM
960 PRINT "TEAM #";TEAM;" -IT'S YOUR TURN": :
970 GOSUB 1190
980 X=INT(RND*125)+1
990 GOSUB 1380
1000 REM
1010 IF X(=25 THEN 1070
1020 IF (X)=26)*(X(=50)THEN 1090
1030 IF (X)=51)*(X(=75)THEN 1110
1040 IF (X)=76)*(X(=100)THEN 1130
1050 PRINT : :"(SONG)": : : :
1060 GOTO 1140
1070 PRINT : :"(MOVIE)": : : :
1080 GOTO 1140
1090 PRINT : :"(BOOK)": : : :
1100 GOTO 1140
1110 PRINT : :"(PERSON)": : : :
1120 GOTO 1140
1130 PRINT : :"(QUOTE OR CLICHE)": : : :
1140 PRINT M$(X): : : : :
1150 GOSUB 1190
1160 PRINT "(HOLD DOWN A KEY TO STOP)"
1170 GOSUB 1280
1180 RETURN
1190 PRINT :P$: :
1200 CALL KEY(0,KEY,STATUS)
1210 IF STATUS=0 THEN 1200
1220 CALL CLEAR
1230 RETURN
1240 CALL SOUND(300,523,2,392,3,330,3)
1250 CALL SOUND(200,494,2,294,3,247,3)
1260 CALL SOUND(400,523,2,392,3,330,3)
1270 RETURN
1280 FOR T=120 TO 1 STEP -1
1290 PRINT T
1300 CALL SOUND(750,40000,30)
1310 CALL SOUND(20,220,10)
1320 CALL KEY(0,KEY,STATUS)
1330 IF STATUS=0 THEN 1340 ELSE 1350
1340 NEXT T
```

```
1350 CALL SOUND(1100,220,0)
1360 PRINT :M$(X): : :TAB(10);"* * * *": : : : :
1370 RETURN
1380 REM  TEST FOR DUP
1390 FOR Y=1 TO Q
1400 IF X=Z(Y)THEN 980
1410 NEXT Y
1420 Z(Q)=X
1430 Q=Q+1
1440 RETURN
1450 REM TUNE
1460 DUR=250
1470 CALL SOUND(DUR,262,1)
1480 CALL SOUND(DUR,277,1)
1490 CALL SOUND(DUR,262,1)
1500 CALL SOUND(DUR*2,392,1)
1510 CALL SOUND(DUR,349,1)
1520 CALL SOUND(DUR*3,262,1)
1530 CALL SOUND(100,40000,30)
1540 CALL SOUND(DUR,262,1)
1550 CALL SOUND(DUR,277,1)
1560 CALL SOUND(DUR,262,1)
1570 CALL SOUND(DUR*2,233,1)
1580 CALL SOUND(DUR,208,1)
1590 CALL SOUND(DUR,262,1)
1600 CALL SOUND(DUR*3,196,1)
1610 RETURN
```

ECHO

This TI BASIC game challenges your visual recognition and memory skills. A series of letters is flashed at various locations on the screen. You must "echo" them back to the computer in the proper order. But don't waste time! The faster you respond, the better your score. It might sound simple, but you may think differently after you select the high level of difficulty. If you make three mistakes, the game ends. As your score increases, you may want to adjust the average scores displayed in line 710.

```
100 REM *ECHO* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 RANDOMIZE
140 DEF A=INT((90-65+1)*RND)+65
150 DEF F=10*A+110
160 DEF R=INT((22-3+1)*RND)+3
170 DEF C=INT((26-5+1)*RND)+5
180 H=40000
190 CALL CLEAR
200 FOR I=3 TO 21
210 PRINT TAB(I);"ECHO"
220 CALL SOUND(-50,300,V)
230 CALL SOUND(-50,250,V)
240 V=V+1
250 NEXT I
260 FOR DELAY=1 TO 150
270 NEXT DELAY
280 INPUT "INSTRUCTIONS? (Y/N):":Y$
290 IF Y$<>"Y" THEN 340
```

```
300 PRINT :"THE COMPUTER WILL FLASH A":"SERIES OF LETTERS ON THE":"SCREEN. YOU M
UST 'ECHO' THEM"
310 PRINT "BACK IN ORDER BY PRESSING":"THE KEYS FOR THOSE LETTERS":"AS QUICKLY A
S POSSIBLE."
320 PRINT "DIFFICULTY INCREASES THE":"LONGER YOU PLAY. THE FASTER":"YOU RESPOND,
 THE BETTER YOUR"
330 PRINT "SCORE. GAME IS OVER WHEN YOU":"MAKE 3 MISTAKES.": : :
340 PRINT "ENTER LEVEL OF DIFFICULTY:":" 1 - LOW":" 2 - HIGH"
350 INPUT L
360 PRINT :"   PRESS ANY KEY TO START"
370 CALL KEY(0,KEY,STATUS)
380 IF STATUS=0 THEN 370
390 CALL CLEAR
400 X=1
410 A$=""
420 FOR I=1 TO X*L
430 Z=A
440 CALL HCHAR(R,C,Z)
450 CALL SOUND(500/L,F,0)
460 CALL SOUND(1,H,30)
470 CALL CLEAR
480 A$=A$&CHR$(Z)
490 NEXT I
500 FOR P=1 TO LEN(A$)
510 CALL KEY(0,KEY,STATUS)
520 T=T+1
530 IF STATUS=0 THEN 510
540 PRINT CHR$(KEY)
550 IF KEY=ASC(SEG$(A$,P,1))THEN 600
560 PRINT "NO,IT WAS ";A$
570 M=M+1
580 IF M>=3 THEN 680
590 GOTO 630
600 NEXT P
610 PRINT "THAT'S RIGHT"
620 X=X+1
630 FOR DELAY=1 TO 400
640 NEXT DELAY
650 CALL CLEAR
660 GOTO 410
670 CALL SOUND(200,110,0)
680 PRINT :"GAME OVER.  AT LEVEL";L;","
690 PRINT "OUT OF";X+2;"SERIES,":"YOU GOT";X-1;"RIGHT"
700 PRINT :"YOU USED";T;"TIME UNITS"
710 PRINT :"SCORE LEVELS:":"  OVER 400  = SUPERIOR":"  150-400   = AVERAGE":"  U
NDER 150 = SLOW"
720 S=(1000/T)*(X+2)*10
730 PRINT :"YOUR SCORE IS";INT(S+1)
740 M=0
750 X=0
760 T=0
770 PRINT :"PLAY AGAIN? (Y/N):"
780 INPUT Y$
790 IF Y$="Y" THEN 340
800 END
```

FOUR-LETTER WORDS

The computer generates 4 random letters that you must guess in this simple TI BASIC game. You have 30 tries to figure out the exact letters and their position in the nonsense four-letter word. At each prompt, you may guess either a single letter or the entire word. See how many turns it takes you to get the word correct. It might be harder than you think.

```
100 REM *4-LETTER WORDS* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM  COPYRIGHT (C) 1983 BY STEVE DAVIS
130 RANDOMIZE
140 CALL CLEAR
150 DISPLAY TAB(6);"FOUR-LETTER WORDS"
160 DISPLAY : : :"THE COMPUTER HAS MADE UP A":"4-LETTER NONSENSE WORD.":"YOU HAV
E 30 TRIES TO GUESS"
170 DISPLAY "IT. EACH TURN, YOU MAY GUESS":"1 LETTER AND THE COMPUTER":"WILL TEL
L YOU HOW MANY TIMES"
180 DISPLAY "IT APPEARS IN THE WORD,":"OR YOU MAY GUESS THE WORD."
190 DISPLAY :"PRESS ANY KEY"
200 CALL KEY(0,KEY,STATUS)
210 ZZ=RND
220 IF STATUS=0 THEN 200
230 CALL CLEAR
240 DEF R=INT(RND*(90-65+1))+65
250 T=0
260 C$=""
270 FOR I=1 TO 4
280 A$(I)=CHR$(R)
290 C$=C$&A$(I)
300 NEXT I
310 DISPLAY "ENTER A SINGLE LETTER OR":"GUESS THE 4-LETTER WORD:": : :
320 INPUT G$
330 IF LEN(G$)=4 THEN 550
340 IF LEN(G$)<>1 THEN 310
350 T=T+1
360 A=0
370 FOR I=1 TO 4
380 IF G$<>A$(I)THEN 400
390 A=A+1
400 NEXT I
410 IF A>1 THEN 470
420 IF A=1 THEN 450
430 DISPLAY "IS NOT IN THE WORD": :
440 GOTO 510
450 DISPLAY "APPEARS 1 TIME!": :
460 GOTO 480
470 DISPLAY "APPEARS";A;"TIMES!": :
480 CALL SOUND(100,660,3)
490 CALL SOUND(100,770,3)
500 CALL SOUND(100,660,3)
510 IF T=30 THEN 680
520 IF T<25 THEN 320
530 DISPLAY :"YOU HAVE";30-T;"TRIES LEFT": :
540 GOTO 320
550 T=T+1
560 IF G$=C$ THEN 650
570 LR=0
580 FOR I=1 TO 4
590 IF SEG$(G$,I,1)<>A$(I)THEN 610
600 LR=LR+1
```

```
610 NEXT I
620 CALL SOUND(150,200,1)
630 DISPLAY "WRONG!";LR;"LETTERS ARE IN":"CORRECT POSITION"
640 GOTO 510
650 CALL SOUND(500,392,1,330,1,262,1)
660 DISPLAY : :"YOU GUESSED IT IN":T;"TRIES!": :
670 GOTO 700
680 CALL SOUND(200,220,1)
690 DISPLAY :"GAME OVER!":"THE WORD WAS ";C$
700 INPUT "TRY AGAIN? (Y/N)":Y$
710 IF Y$="Y" THEN 230
720 STOP
```

LUCKY SEVEN

Try your luck against the dealer in this TI BASIC card game. You may place a bet and draw from one to seven cards. Your score is the sum of the value of each card you draw (Ace=1; Jack, Queen, King=10). However, if you draw a card like the one you first drew, then you "bust" and lose your bet. For example, if you drew a Jack, a 5 and an Ace, your score would be 16. Then, if you drew another Jack you would bust. If you have not busted before you end your turn, then the computer tries to beat your score. If he busts, then you win! If you have drawn seven cards, then you win double your bet! In case of a tie, you neither win nor lose any money.

```
100 REM *LUCKY SEVEN* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL SCREEN(16)
140 DIM C$(52)
150 RANDOMIZE
160 DEF CN=VAL(SEG$(C$(Q),1,3))
170 DEF CS=VAL(SEG$(C$(Q),4,3))
180 CALL CHAR(64,"0038545038145438")
190 CALL CHAR(62,"0008081000000000")
200 CALL CHAR(60,"0010101010100010")
210 CALL CLEAR
220 CALL HCHAR(1,1,55,768)
230 A$="  LUCKY  SEVEN   "
240 FOR I=1 TO 16
250 CALL VCHAR(1,7+I,ASC(SEG$(A$,I,1)),22)
260 CALL SOUND(-25,220+I*50,3)
270 NEXT I
280 CALL HCHAR(23,8,32,16)
290 GOSUB 500
300 CALL CLEAR
310 INPUT "INSTRUCTIONS? (Y/N)":Y$
320 IF Y$()"Y" THEN 400
330 PRINT : : :"PLACE A BET, DRAW A CARD.":"YOUR SCORE IS THE COMBINED  VALUE OF
 CARDS YOU DRAW"
340 PRINT "(A=1; J,Q,K=10). DRAW UP TO 7 CARDS, BUT WATCH OUT!":"IF YOU DRAW A C
ARD LIKE THE ONE YOU FIRST DREW"
350 PRINT "THEN YOU 'BUST' & LOSE":"DOUBLE YOUR BET. YOU MAY":"STOP DRAWING AT A
NYTIME.":"COMPUTER TAKES A TURN"
360 PRINT "& IF HE GETS A HIGHER SCORE":"WITHOUT BUSTING, HE WINS,":"OTHERWISE Y
OU WIN."
370 PRINT "IF YOU DREW 7 CARDS AND WIN,THEN YOU WIN DOUBLE!":"(TIP: A STRATEGY I
S TO":"REMEMBER WHAT CARDS HAVE"
```

```
380 PRINT "BEEN PLAYED.)":"GOOD LUCK!": :
390 GOSUB 500
400 PRINT :"HOW MUCH MONEY DO YOU"
410 INPUT "HAVE? $":M
420 PRINT :"INITIALIZING...": :
430 M=INT(M+.5)
440 GOSUB 580
450 A$="SHUFFLING CARDS"
460 GOSUB 900
470 GOSUB 960
480 CALL SCREEN(13)
490 GOTO 1120
500 CALL HCHAR(24,1,32,32)
510 A$=" PRESS A KEY "
520 FOR I=1 TO 13
530 CALL HCHAR(24,I+8,ASC(SEG$(A$,I,1)))
540 NEXT I
550 CALL KEY(0,KEY,STATUS)
560 IF STATUS=0 THEN 550
570 RETURN
580 REM  DEF CARD CHARACTERS AND VALUES
590 FOR I=104 TO 116
600 READ A$
610 CALL CHAR(I,A$)
620 CALL CHAR(I+16,A$)
630 NEXT I
640 DATA 003844447C444444,0038440408102070,003844041804443 8,00081828487C0808
650 DATA 007C407804044438,0018204078444438,007C040810202020,0038444438444438
660 DATA 003844443C040830,005E5252525252 5E,0004040404044438,0038444444544834
670 DATA 0044485060504844
680 DATA 01,02,03,04,05,06,07,08,09,10,10,10,10
690 CALL COLOR(10,2,16)
700 CALL COLOR(11,2,16)
710 CALL COLOR(12,9,16)
720 CALL COLOR(13,9,16)
730 REM DEFINE CARD SUITS
740 CALL CHAR(133,"0066E7FFFF7E3C18")
750 CALL CHAR(134,"00183C7EFF7E3C18")
760 CALL CHAR(117,"3C3C18DBFFDB1818")
770 CALL CHAR(118,"183C7EFFFFFFDB18")
780 FOR I=1 TO 13
790 READ V$
800 C$(I)=STR$(I+103)&STR$(117)&V$
810 C$(I+13)=STR$(I+103)&STR$(118)&V$
820 C$(I+26)=STR$(I+119)&STR$(133)&V$
830 C$(I+39)=STR$(I+119)&STR$(134)&V$
840 NEXT I
850 CALL CHAR(96,"8080808080808080")
860 CALL CHAR(97,"0000000000000000")
870 CALL CHAR(92,"0008081000000000")
880 CALL COLOR(9,3,16)
890 RETURN
900 T=INT(32-LEN(A$))/2
910 CALL HCHAR(24,1,32,32)
920 FOR I=1 TO LEN(A$)
930 CALL HCHAR(24,T+I,ASC(SEG$(A$,I,1)))
940 NEXT I
950 RETURN
960 FOR I=1 TO 52
970 CALL SOUND(-2,-5,6)
980 X=INT(RND*52)+1
990 X$=C$(I)
1000 C$(I)=C$(X)
1010 C$(X)=X$
1020 NEXT I
```

```
1030 CALL SOUND(100,600,3)
1040 Q=0
1050 RETURN
1060 CALL CLEAR
1070 PRINT :"HOW MUCH MONEY DO YOU"
1080 INPUT "HAVE? $":M
1090 M=INT(M+.5)
1100 SCR=0
1110 IF M<=0 THEN 2380
1120 CALL CLEAR
1130 IF Q<39 THEN 1170
1140 A$="SHUFFLING CARDS"
1150 GOSUB 900
1160 GOSUB 960
1170 MC=0
1180 CALL CLEAR
1190 PRINT "YOU HAVE @"&STR$(M): :
1200 INPUT "HOW MUCH DO YOU BET? @":B
1210 B=INT(B+.5)
1220 IF B<=M THEN 1250
1230 DISPLAY "YOU DON'T HAVE THAT MUCH"
1240 GOTO 1200
1250 CALL CLEAR
1260 F=1
1270 GOSUB 1290
1280 GOTO 1330
1290 FOR I=3 TO 8
1300 CALL COLOR(I,F,1)
1310 NEXT I
1320 RETURN
1330 PRINT "YOU HAVE @"&STR$(M);TAB(17);"YOU BET @"&STR$(B): :TAB(7);"YOUR SCORE
: 00": : : : : : : : : : : : : : : : : :
1340 MC=0
1350 PRINT "PRESS D TO DRAW OR Q TO QUIT"
1360 BONUS=0
1370 F=16
1380 GOSUB 1290
1390 CALL KEY(0,KEY,STATUS)
1400 IF KEY=81 THEN 1750
1410 IF (STATUS=0)+(KEY<>68)THEN 1390
1420 MC=MC+1
1430 Q=Q+1
1440 IF MC>1 THEN 1460
1450 FC=CN
1460 GOSUB 1480
1470 GOTO 1570
1480 CALL VCHAR(8,2+MC*3,96,10)
1490 FOR I=1 TO 6
1500 CALL VCHAR(8,2+I+MC*3,97,10)
1510 NEXT I
1520 CALL HCHAR(9,3+MC*3,CN)
1530 CALL HCHAR(10,3+MC*3,CS)
1540 CALL HCHAR(15,7+MC*3,CN)
1550 CALL HCHAR(16,7+MC*3,CS)
1560 RETURN
1570 IF MC=1 THEN 1640
1580 IF (CN>FC)*(CN<>FC+16)THEN 1640
1590 IF (CN<FC)*(CN<>FC-16)THEN 1640
1600 GOTO 2290
1610 CALL SOUND(200,110,0,220,0)
1620 GOSUB 2350
1630 IF Q<39 THEN 1120 ELSE 450
1640 V=VAL(SEG$(C$(Q),7,2))
1650 SCR=SCR+V
1660 SC$=STR$(SCR)
```

```
1670 IF SCR>9 THEN 1690
1680 SC$="0"&SC$
1690 CALL HCHAR(4,21,ASC(SEG$(SC$,1,1)))
1700 CALL HCHAR(4,22,ASC(SEG$(SC$,2,1)))
1710 CALL SOUND(50,600,3)
1720 IF MC<7 THEN 1390
1730 BONUS=1
1740 GOSUB 2350
1750 CALL CLEAR
1760 F=1
1770 GOSUB 1290
1780 PRINT "YOU HAVE @"&STR$(M);TAB(17);"YOU BET @"&STR$(B): :TAB(7);"YOUR SCORE
:";SCR: : : : : : : : : : : : : : :
1790 PRINT TAB(7);"COMPUTER>S TURN": :TAB(5);"COMPUTER>S SCORE: 00"
1800 F=16
1810 GOSUB 1290
1820 MC=0
1830 CSC=0
1840 MC=MC+1
1850 Q=Q+1
1860 IF MC>1 THEN 1880
1870 FC=CN
1880 GOSUB 1480
1890 IF MC=1 THEN 1960
1900 IF (CN>FC)*(CN<>FC+16)THEN 1960
1910 IF (CN<FC)*(CN<>FC-16)THEN 1960
1920 A$="COMPUTER BUSTED< YOU WIN<"
1930 GOSUB 900
1940 CALL SOUND(400,262,3,330,5,392,5)
1950 GOTO 2130
1960 V=VAL(SEG$(C$(Q),7,2))
1970 CSC=CSC+V
1980 CSC$=STR$(CSC)
1990 IF CSC>9 THEN 2010
2000 CSC$="0"&CSC$
2010 CALL HCHAR(23,25,ASC(SEG$(CSC$,1,1)))
2020 CALL HCHAR(23,26,ASC(SEG$(CSC$,2,1)))
2030 CALL SOUND(50,600,3)
2040 IF CSC>SCR THEN 2230
2050 IF MC<7 THEN 1840
2060 IF CSC<>SCR THEN 2120
2070 A$="A TIE< TRY AGAIN"
2080 CALL SOUND(300,440,3)
2090 GOSUB 900
2100 GOSUB 2350
2110 GOTO 1100
2120 IF CSC>SCR THEN 2230
2130 IF BONUS=0 THEN 2170
2140 A$="YOU WIN DOUBLE<"
2150 B=B*2
2160 GOTO 2180
2170 A$="YOU WIN<"
2180 CALL SOUND(400,262,3,330,5,392,5)
2190 GOSUB 900
2200 GOSUB 2350
2210 M=M+B
2220 GOTO 1100
2230 A$="COMPUTER WINS<"
2240 CALL SOUND(200,110,0,220,0)
2250 GOSUB 900
2260 M=M-B
2270 GOSUB 2350
2280 GOTO 1100
2290 A$="YOU BUSTED<"
2300 CALL SOUND(200,110,0,220,0)
2310 M=M-(2*B)
```

```
2320 GOSUB 900
2330 GOSUB 2350
2340 GOTO 1100
2350 FOR D=1 TO 350
2360 NEXT D
2370 RETURN
2380 CALL CLEAR
2390 F=2
2400 GOSUB 1290
2410 PRINT "YOU HAVE NO MONEY LEFT!!"
2420 INPUT "WANT TO PLAY AGAIN? (Y/N)":Y$
2430 IF Y$<>"Y" THEN 2450
2440 GOTO 1060
2450 PRINT : : :"BETTER LUCK NEXT TIME!"
```

MOVIE STAR QUIZ

Here is a game for trivia buffs. This quiz game contains 30 questions about movie stars which can be answered in 3 rounds, with 10 questions in each round. For each question, 3 answers are displayed, and you must choose which one is correct. Each time you play, the correct answer may be in a different position, and the wrong answers will not be the same each time. Also, the order in which you receive the questions varies from game to game.

It is a good idea to let a friend type in the Data statements so that you will not be familiar with the questions and answers before you play. This program could very easily be adapted to quiz you on practically any kind of trivia by changing the data.

The random number (X) selects which question will be asked next when used as a subscript to A$ and Q$. The variable Q is a counter which keeps track of how many questions have been asked. The variable N(Q) represents the actual question numbers that have been used. Thus, each time the random X is generated, it is checked against the numbers already used. If it has been previously used in the game, then another number is generated. Because of this process, no question appears twice in the same game, but it will naturally take a little longer for the computer to select an unused question toward the end of the game.

Random numbers W, Y, and Z are used to determine which wrong answers will be used and in which order they will be printed. When you have used up the 30 questions, the game ends.

Teachers might want to use this program as a model for a self-administering multiple choice test. By using the "PROTECTED" option when saving the program in TI Extended BASIC, the user would be prevented from listing the program data and seeing the questions and answers.

```
100 REM *MOVIE STAR QUIZ* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 OPTION BASE 1
140 DIM A$(30),Q$(30),N(30)
150 CALL CLEAR
160 C$="MOVIE STAR"
170 B$="   QUIZ   "
180 FOR I=1 TO 10
190 CALL HCHAR(11,10+I,ASC(SEG$(C$,I,1)))
200 CALL HCHAR(13,10+I,ASC(SEG$(B$,I,1)))
210 NEXT I
220 GOSUB 1190
230 GOSUB 820
240 FOR I=1 TO 30
250 READ Q$(I),A$(I)
260 NEXT I
270 Q=1
280 RANDOMIZE
290 FOR I=1 TO 10
300 CALL CLEAR
310 GOSUB 750
320 PRINT "WHICH STAR ";Q$(X);"?"
330 Y=INT(RND*3)+1
340 W=INT(RND*30)+1
350 IF W=X THEN 340
360 Z=INT(RND*30)+1
370 IF (Z=X)+(Z=W)THEN 360
380 IF Y<>1 THEN 410
390 PRINT :"1. ";A$(X):"2. ";A$(W):"3. ";A$(Z)
400 GOTO 450
410 IF Y<>2 THEN 440
420 PRINT :"1. ";A$(W):"2. ";A$(X):"3. ";A$(Z)
430 GOTO 450
440 PRINT :"1. ";A$(W):"2. ";A$(Z):"3. ";A$(X)
450 PRINT
460 INPUT C
470 PRINT
480 IF C<>Y THEN 560
490 F=110
500 FOR J=3 TO 8
510 CALL SOUND(100,J*F,1)
520 NEXT J
530 S=S+1
540 PRINT "THAT'S RIGHT!": :
550 GOTO 580
560 CALL SOUND(100,110,1)
570 PRINT "SORRY--TRY THE NEXT QUESTION"
580 FOR DELAY=1 TO 300
590 NEXT DELAY
600 NEXT I
610 PRINT :"YOUR SCORE IS:";S*10
620 IF S>6 THEN 650
630 PRINT "BETTER LUCK NEXT TIME"
640 GOTO 700
650 F=110
660 FOR J=2 TO 8
```

```
670 CALL SOUND(200,J*F,1,(J-1)*F,2)
680 NEXT J
690 PRINT "VERY GOOD"
700 IF Q>29 THEN 740
710 S=0
720 INPUT "PLAY AGAIN(Y/N)?";Y$
730 IF Y$="Y" THEN 280
740 END
750 X=INT(RND*30)+1
760 FOR K=1 TO Q
770 IF X=N(K)THEN 750
780 NEXT K
790 N(Q)=X
800 Q=Q+1
810 RETURN
820 CALL SOUND(300,392,1)
830 CALL SOUND(300,349,1)
840 CALL SOUND(300,333,1)
850 CALL SOUND(500,392,1)
860 CALL SOUND(300,196,1)
870 CALL SOUND(700,220,1)
880 RETURN
890 DATA WON AN OSCAR FOR 'THE BAD & THE BEAUTIFUL',GLORIA GRAHAME
900 DATA HAD ONE ARM IN 'BAD DAY AT  BLACK ROCK',SPENCER TRACY
910 DATA WAS THE VOICE FOR MR. MAGOO,JIM BACKUS
920 DATA HAD A SMALL ROLE IN 'THE    ASPHALT JUNGLE',MARILYN MONROE
930 DATA GOT AN OSCAR NOMINATION FOR A SUPPORTING ROLE IN 'SINGIN IN THE RAIN',J
EAN HAGEN
940 DATA PLAYED NORMAN BATES IN      'PSYCHO',ANTHONY PERKINS
950 DATA PLAYED OPPOSITE MARILYN     MONROE IN 'THE MISFITS',CLARK GABLE
960 DATA WON AN OSCAR FOR HITCHCOCK'S 'SUSPICION',JOAN FONTAINE
970 DATA PORTRAYED 'THE GREAT        ZIEGFELD',WILLIAM POWELL
980 DATA PLAYED BLANCHE HUDSON,JOAN CRAWFORD
990 DATA HAD BREAKFAST AT TIFFANY'S,AUDREY HEPBURN
1000 DATA GOT STABBED IN 'PSYCHO',MARTIN BALSAM
1010 DATA WAS 'THE MIRACLE WORKER',ANNE BANCROFT
1020 DATA SANG 'SAN FRANCISCO',JEANETTE MACDONALD
1030 DATA DID GLORIA SWANSON SHOOT IN 'SUNSET BLVD',WILLIAM HOLDEN
1040 DATA SAID 'I DETEST CHEAP       SENTIMENT',BETTE DAVIS
1050 DATA PLAYED ARMAND IN 'CAMILLE',ROBERT TAYLOR
1060 DATA SANG ABOUT CLARK GABLE,JUDY GARLAND
1070 DATA PLAYED DOCTOR HACKENBUSH?,GROUCHO MARX
1080 DATA SLAPPED JOAN CRAWFORD IN   'MILDRED PIERCE',ANN BLYTH
1090 DATA KNOWS HOW TO DO THE CARIOCA,FRED ASTAIRE
1100 DATA SAID 'I GENERALLY AVOID    TEMPTATION UNLESS I CAN'T   RESIST IT',MAE
 WEST
1110 DATA SAID 'I WENT TO PHILADELPHIAONE TIME; IT WAS CLOSED',WC FIELDS
1120 DATA KICKED THE BUCKET IN 'ITS A MAD MAD WORLD',JIMMIE DURANTE
1130 DATA PORTRAYED DOCTOR GILLESPIE,LIONEL BARRYMORE
1140 DATA WAS CALLED 'STONE FACE',BUSTER KEATON
1150 DATA SUFFERED FROM VERTIGO,JAMES STEWART
1160 DATA HAD A TOOTHACHE IN 'GREED',ZASU PITTS
1170 DATA HAD THEIR SINGING VOICE    DUBBED BY ANDY WILLIAMS,LAUREN BACALL
1180 DATA MADE A COMEBACK IN 'STATE  FAIR' IN 1961,ALICE FAYE
1190 REM  FLASHING MARQUEE
1200 CALL CHAR(96,"FFC381818181C3FF")
1210 CALL SCREEN(15)
1220 CALL HCHAR(1,1,96,128)
1230 CALL HCHAR(21,1,96,128)
1240 CALL VCHAR(1,1,96,96)
1250 CALL VCHAR(1,29,96,96)
1260 FOR I=1 TO 8
1270 CALL COLOR(9,2,16)
1280 FOR D=1 TO 50
1290 NEXT D
```

```
1300 CALL COLOR(9,16,2)
1310 FOR D=1 TO 50
1320 NEXT D
1330 NEXT I
1340 RETURN
```

MURDER

You are in a beautiful Victorian mansion with 12 rooms. There are a dozen other people there, including servants and a number of dignitaries. After a delicious meal, you decide it is time to leave. But, before you can get out the door, your help is summoned. Can it be foul play? Yes, it is definitely a case of . . . MURDER!

Now, it is your job, as a famous private investigator, to find out who has been done in. Where was it done? Who dunit? And, how? See how quickly you can deduce the answer. Pay attention, because each game is a little different, and a few surprises are always possible in these types of cases. You must find the body of the victim in one of the rooms to know who suffered this horrible deed. What if the murder suspect and the victim are one and the same? "Impossible," you say? What if it was suicide?

You can only make one move or one guess in a turn, and you can only guess weapons or suspects that are present in the room you are in. You only get one chance to make a complete accusation, so be sure you have all the clues before you do. If your accusation is correct, you win. Otherwise, the computer has given you a case too difficult to solve.

At the beginning of the program, X is defined as a random number. Each time you play, the suspects and weapons are placed randomly in various rooms throughout the house. If a suspect appears as just a body, obviously he is the victim, S$(V). The suspect is represented by S$(S), and R$(R) is the room, while W$(W) is the weapon. The location of each is noted by SLOC(suspect number) and WLOC (weapon number). Your present location is held in the variable PLOC.

The computer will count a turn each time you move or make a guess. If you give up or enter a false accusation, the computer will tell you who dunit. The listing is in TI BASIC, but you may find it runs slightly faster in the latest version of TI Extended BASIC. There are over 10,000 possible solutions to the crime, so the game should provide any aspiring super-sleuth with many hours of deductive fun.

```
100 REM *MURDER*  TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 RANDOMIZE
140 DEF X=INT(RND*12)+1
150 OPTION BASE 1
160 DIM R$(12),S$(12),W$(6),SLOC(12),WLOC(12)
170 FOR I=1 TO 12
```

```
180 READ R$(I),S$(I)
190 IF I)6 THEN 210
200 READ W$(I)
210 NEXT I
220 DATA KITCHEN,COOK,KNIFE,GREENHOUSE,GARDENER,SHOVEL,GARAGE,CHAUFFEUR,ROPE
230 DATA LIVING ROOM,DUKE,POISON,ATTIC,DUCHESS,GUN,BASEMENT,COUNT,HAMMER
240 DATA STUDY,COUNTESS,BATH,BUTLER,HALL,NANNY,DEN,MAYOR,BEDROOM,OPERA STAR,DINI
NG ROOM,GOVERNOR
250 CALL CLEAR
260 PRINT "YOU ARE A PRIVATE DETECTIVE AT A FASHIONABLE DINNER     PARTY"
270 PRINT : :"SEVERAL DISTINGUISHED GUESTSARE PRESENT AS WELL AS ALL  THE SERVAN
TS"
280 FOR I=1 TO 800
290 NEXT I
300 PRINT : :"SUDDENLY...": : :
310 FOR I=1 TO 200
320 NEXT I
330 CALL SCREEN(7)
340 PRINT : :TAB(12);"MURDER!": : : :
350 CALL SOUND(500,250,0)
360 CALL SOUND(500,250,0)
370 CALL SOUND(200,250,0)
380 CALL SOUND(500,250,0)
390 CALL SOUND(500,300,0)
400 CALL SOUND(200,280,0)
410 CALL SOUND(500,280,0)
420 CALL SOUND(200,250,0)
430 CALL SOUND(500,250,0)
440 CALL SOUND(200,240,0)
450 CALL SOUND(500,250,0)
460 CALL CLEAR
470 CALL SCREEN(8)
480 PRINT "WHO GOT KILLED??": :"WAS IT...": : :
490 FOR I=1 TO 12
500 PRINT "THE ";S$(I);"?"
510 NEXT I
520 GOSUB 800
530 PRINT "WHO COMMITTED THE MURDER??": :"COULD IT HAVE BEEN...": : :
540 FOR I=12 TO 1 STEP -1
550 PRINT "THE ";S$(I);"?"
560 NEXT I
570 GOSUB 800
580 PRINT "WHERE DID THEY DO IT??": :"WAS IT IN...": : :
590 FOR I=1 TO 12
600 PRINT "THE ";R$(I);"?"
610 NEXT I
620 GOSUB 800
630 PRINT "AND HOW DID THEY DO IT??": :"DID THEY USE...": : :
640 FOR I=1 TO 6
650 PRINT "THE ";W$(I);"?"
660 NEXT I
670 PRINT : :"IT'S UP TO YOU TO DECIDE!": :"(BUT BE CAREFUL BECAUSE LIFE IS FULL
 OF SURPRISES!)"
680 GOSUB 800
690 PRINT "YOU MAY GUESS ONLY 1 ITEM INA TURN (ROOM,SUSPECT OR":"WEAPON) & ONLY
THE ROOM OR  ITEMS IN THE ROOM YOU ARE IN"
700 PRINT :"OR YOU MAY USE A TURN TO":"MOVE TO ANOTHER ROOM": :"OR YOU MAY USE A
 TURN TO    MAKE AN ACCUSATION, NAMING"
710 PRINT "THE VICTIM,ROOM,MURDERER ANDWEAPON. BUT YOU CAN MAKE ONEACCUSATION ON
LY."
720 PRINT "IF IT IS COMPLETELY CORRECT,YOU WIN. IF NOT, THE GAME ISOVER & I WIN.
 GOOD LUCK!"
730 GOSUB 800
740 INPUT "ENTER YOUR NAME:":NAM$
750 S=X
```

```
760 V=X
770 R=X
780 W=INT(RND*6)+1
790 GOTO 850
800 PRINT : :"(PRESS ANY KEY TO CONTINUE)"
810 CALL KEY(0,KEY,STATUS)
820 IF STATUS=0 THEN 810
830 CALL CLEAR
840 RETURN
850 FOR I=1 TO 12
860 SLOC(I)=X
870 WLOC(I)=X
880 NEXT I
890 TURN=0
900 PLOC=X
910 PRINT : : : : : : :NAM$;", YOU ARE IN THE":R$(PLOC)
920 GOTO 1120
930 TURN=TURN+1
940 GOTO 910
950 PRINT :"IT TOOK YOU";TURN+1;"TURNS":"TO SOLVE THIS MYSTERY"
960 PRINT :"PLAY AGAIN? (Y/N):"
970 INPUT Y$
980 IF Y$<>"Y" THEN 1010
990 CALL CLEAR
1000 GOTO 750
1010 STOP
1020 FOR J=1 TO 12
1030 PRINT J;TAB(5);"- ";R$(J)
1040 NEXT J
1050 PRINT :"WHICH ROOM?"
1060 INPUT A
1070 IF (A<>INT(A))+(A<1)+(A>12)THEN 1050
1080 IF A=PLOC THEN 1110
1090 PLOC=A
1100 GOTO 930
1110 PRINT "YOU'RE ALREADY THERE"
1120 PRINT :"THESE SUSPECTS ARE PRESENT:"
1130 FOR J=1 TO 12
1140 IF SLOC(J)<>PLOC THEN 1190
1150 IF V<>J THEN 1180
1160 PRINT "  THE BODY OF THE":"    ";S$(V)
1170 GOTO 1190
1180 PRINT "  ";S$(J)
1190 NEXT J
1200 PRINT :"THESE WEAPONS ARE PRESENT:"
1210 FOR J=1 TO 6
1220 IF WLOC(J)<>PLOC THEN 1240
1230 PRINT "  ";W$(J)
1240 NEXT J
1250 PRINT :"ENTER":"1 - TO GUESS THIS ROOM":"2 - TO GUESS A SUSPECT IN":"    TH
IS ROOM"
1260 PRINT "3 - TO GUESS A WEAPON IN":"    THIS ROOM":"4 - TO MAKE AN ACCUSATION
":"5 - TO MOVE TO ANOTHER ROOM"
1270 INPUT A
1280 IF (A<1)+(A>5)THEN 1270
1290 ON A GOTO 1300,1390,1610,1900,1020
1300 IF R=PLOC THEN 1350
1310 PRINT :"NO, IT WAS NOT DONE IN THE":R$(PLOC)
1320 GOSUB 1870
1330 GOSUB 800
1340 GOTO 930
1350 PRINT "YES, IT WAS DONE IN THE":R$(R)
1360 GOSUB 1830
1370 GOSUB 800
1380 GOTO 930
```

```
1390 PRINT
1400 FOR J=1 TO 12
1410 IF SLOC(J)<>PLOC THEN 1430
1420 PRINT S$(J)
1430 NEXT J
1440 PRINT :"ENTER NAME OF THE SUSPECT:"
1450 INPUT A$
1460 FOR J=1 TO 12
1470 IF A$<>S$(J)THEN 1500
1480 IF SLOC(J)<>PLOC THEN 1510
1490 IF A$=S$(S)THEN 1570 ELSE 1530
1500 NEXT J
1510 PRINT "THE ";A$;" IS NOT":"IN THIS ROOM"
1520 GOTO 910
1530 PRINT "NO, IT WASN'T THE":A$
1540 GOSUB 1870
1550 GOSUB 800
1560 GOTO 930
1570 PRINT "YES, IT WAS THE":S$(S)
1580 GOSUB 1830
1590 GOSUB 800
1600 GOTO 930
1610 PRINT
1620 FOR J=1 TO 6
1630 IF WLOC(J)<>PLOC THEN 1650
1640 PRINT W$(J)
1650 NEXT J
1660 PRINT :"ENTER THE NAME OF THE":"WEAPON:"
1670 INPUT A$
1680 FOR J=1 TO 6
1690 IF A$<>W$(J)THEN 1720
1700 IF WLOC(J)<>PLOC THEN 1730
1710 IF A$=W$(W)THEN 1750 ELSE 1790
1720 NEXT J
1730 PRINT "THE ";A$;" IS NOT IN":"THIS ROOM"
1740 GOTO 910
1750 PRINT "YES, IT WAS DONE WITH THE":W$(W)
1760 GOSUB 1830
1770 GOSUB 800
1780 GOTO 930
1790 PRINT "NO, IT WASN'T THE":A$
1800 GOSUB 1870
1810 GOSUB 800
1820 GOTO 930
1830 FOR SND=1 TO 6
1840 CALL SOUND(50,330+SND*50,0)
1850 NEXT SND
1860 RETURN
1870 CALL SOUND(100,240,0)
1880 CALL SOUND(200,130,0)
1890 RETURN
1900 PRINT :"ARE YOU SURE YOU KNOW THE":"VICTIM,ROOM,MURDERER &":"WEAPON? (Y/N):
"
1910 INPUT Y$
1920 IF Y$<>"Y" THEN 1250
1930 PRINT :
1940 FOR J=1 TO 12
1950 PRINT J;TAB(5);"- ";S$(J)
1960 NEXT J
1970 E$="(ENTER NUMBER)"
1980 PRINT :"WHO WAS THE VICTIM?":E$
1990 INPUT V1
2000 PRINT :
2010 FOR J=1 TO 12
2020 PRINT J;TAB(5);"- ";R$(J)
```

```
2030 NEXT J
2040 PRINT :"WHICH ROOM?":E$
2050 INPUT R1
2060 PRINT :
2070 FOR J=1 TO 12
2080 PRINT J;TAB(5);"- ";S$(J)
2090 NEXT J
2100 PRINT :"WHO WAS THE MURDERER?":E$
2110 INPUT S1
2120 PRINT :
2130 FOR J=1 TO 6
2140 PRINT J;TAB(5);"- ";W$(J)
2150 NEXT J
2160 PRINT :"WHICH WEAPON?":E$
2170 INPUT W1
2180 IF (V1=V)*(R1=R)*(S1=S)*(W1=W)THEN 2260
2190 GOSUB 1870
2200 PRINT :"I'M SORRY, BUT THAT IS NOT":"ENTIRELY CORRECT!"
2210 PRINT :"THE ";S$(V);" WAS KILLED":"IN THE ";R$(R);" BY THE":S$(S);" WITH TH
E ";W$(W)
2220 IF V<>S THEN 2240
2230 PRINT :"SUICIDE! HOW TRAGIC!"
2240 PRINT :"WELL I GUESS I FOOLED YOU   THIS TIME!"
2250 GOTO 960
2260 PRINT :"YES, THAT IS RIGHT!":"YOU'RE SUCH A CLEVER SLEUTH!"
2270 GOSUB 1830
2280 GOTO 950
```

TEN-UP

In this TI BASIC game, you are dealt ten cards from a deck of 50 cards which are numbered from 1 to 50. The values of your cards are displayed on the screen. The object is to arrange your cards in ascending order in as few turns as possible. When finished, the lowest card of your hand should be at the top of the screen and the highest card at the bottom. In each turn, you may draw a card and decide whether to exchange it for one in your hand or to discard it to the bottom of the deck. If you can complete the game in less than 10 turns, you are doing well.

```
100 REM *TEN-UP* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM  COPYRIGHT (C) 1983 BY STEVE DAVIS
130 RANDOMIZE
140 DIM DR(50),MC(10)
150 CALL CHAR(96,"00181818DB7E3C18")
160 CALL CLEAR
170 PRINT TAB(11);"TEN-UP"
180 PRINT :"THERE ARE 50 NUMBERED CARDS IN THE DECK. SEE HOW MANY   TURNS IT TAK
ES YOU TO PUT"
190 PRINT "YOUR 10 CARDS IN ORDER FROM LOW TO HIGH."
200 PRINT :"IN YOUR TURN YOU CAN DRAW   A CARD AND EITHER EXCHANGE  IT FOR ONE O
F YOUR CARDS OR DISCARD IT TO THE BOTTOM OF THE DECK."
210 PRINT :"PRESS ANY KEY TO CONTINUE"
220 CALL KEY(0,KEY,STATUS)
230 Z=RND
240 IF STATUS=0 THEN 220
250 PRINT :TAB(5);"PLEASE WAIT WHILE I":TAB(6);"SHUFFLE THE CARDS"
260 TURN=0
270 FOR I=1 TO 50
```

```
280 DR(I)=I
290 NEXT I
300 FOR I=1 TO 50
310 CALL SOUND(-2,-5,6)
320 R=INT(RND*50)+1
330 X=DR(I)
340 DR(I)=DR(R)
350 DR(R)=X
360 NEXT I
370 FOR I=1 TO 10
380 MC(I)=DR(I+40)
390 NEXT I
400 CALL CLEAR
410 PRINT "PUT YOUR CARDS IN ORDER FROMLOW TO HIGH, TOP TO BOTTOM."
420 PRINT :"HERE ARE YOUR CARDS:":"(LOW)"
430 FOR I=1 TO 10
440 PRINT "  ";CHR$(96);TAB(6);STR$(MC(I))
450 NEXT I
460 PRINT "(HIGH)"
470 PRINT :"PRESS D TO DRAW"
480 CALL KEY(0,KEY,STATUS)
490 IF STATUS=0 THEN 480
500 IF KEY=68 THEN 520
510 GOTO 480
520 TC=DR(1)
530 CALL SOUND(50,660,3)
540 PRINT :"YOU DREW";TC:"PRESS K TO KEEP CARD":"PRESS X TO DISCARD IT"
550 CALL KEY(0,KEY,STATUS)
560 IF STATUS=0 THEN 550
570 IF KEY=75 THEN 640
580 IF KEY<>88 THEN 550
590 FOR I=1 TO 39
600 DR(I)=DR(I+1)
610 NEXT I
620 DR(40)=TC
630 GOTO 740
640 INPUT "DISCARD WHICH CARD?":N
650 FOR I=1 TO 10
660 IF MC(I)=N THEN 700
670 NEXT I
680 PRINT "YOU DON'T HAVE THAT CARD!"
690 GOTO 640
700 X=MC(I)
710 MC(I)=TC
720 TC=X
730 GOTO 590
740 TURN=TURN+1
750 FOR I=1 TO 9
760 IF MC(I)<MC(I+1)THEN 770 ELSE 870
770 NEXT I
780 PRINT :"IT TOOK YOU";TURN;"TURNS!"
790 IF TURN>10 THEN 820
800 PRINT "THAT'S PRETTY GOOD!"
810 GOTO 830
820 PRINT "SEE IF YOU CAN DO BETTER":"NEXT TIME."
830 CALL SOUND(1000,262,0,330,0,392,0)
840 INPUT "PLAY AGAIN? (Y/N)":Y$
850 IF Y$="Y" THEN 250
860 STOP
870 CALL CLEAR
880 GOTO 420
```

TI KENO

One reason Keno is such a popular game in the casinos of Las Vegas and Reno is the large jackpots offered, usually as high as $50,000. It is a simple game to play because all you have to do is select 8 numbers between 1 and 80. The house selects 20 numbers in the same range, and if your numbers match, you win. The casinos make a great deal of money on this game because of its deceptive simplicity.

Actually, the odds of you choosing 8 numbers that will be chosen by the house are very low, and that is why such large jackpots can be offered. In this computer version of Keno, you will find that it may take a number of games for you to win a sizeable amount. The amount you win depends on how many of the numbers you match and the amount of your bet. Also, in this version, unlike in Las Vegas, there is no limit to the amount you can win. You can win up to 2000 times your bet if all your numbers match, but if you do not match at least 5 numbers, you do not win anything.

When you are asked to enter the 8 numbers of your choosing, there are check routines included in the program to be sure that no duplicate numbers are entered or that no number is a non-integer or that it is between 1 and 80 inclusive. The computer chooses its 20 numbers at random, checking its own numbers for possible duplicates, then it compares its numbers with yours and tells you how many matched and what your winnings are, if any. When asked if you wish to play again, the CALL KEY routine, which waits for your input, includes a statement, Z=RND (line 910), that generates random numbers. This assures that you will be at a different point in the computer's pseudo-random number generator each time you play and be using truly unpredictable numbers.

This TI BASIC casino-style game of chance may start you thinking of ways to beat the system, but after awhile, you may just be glad that it is only play money.

```
100 REM *TI KENO* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM YN(8),CN(20)
140 RANDOMIZE
150 GOSUB 1270
160 PRINT TAB(10);"TI KENO"
170 PRINT : :"CHOOSE 8 DIFFERENT NUMBERS":"FROM 1 TO 80. COMPUTER PICKS":"20 NUM
BERS. IF YOU MATCH 5"
180 PRINT "OR MORE NUMBERS, YOU WIN!":"PAYOFF:":" MATCH 5 = WIN 10 TIMES BET"
190 PRINT " MATCH 6 = WIN 100 TIMES":TAB(16);"YOUR BET"
200 PRINT " MATCH 7 = WIN 2000 TIMES":TAB(16);"YOUR BET":" MATCH 8 = WIN 20000 T
IMES":TAB(16);"YOUR BET"
210 PRINT :"HOW MUCH MONEY DO YOU HAVE  TO PLAY WITH?"
220 INPUT "$":T
230 T=INT(T*100)/100
240 CALL CLEAR
250 PRINT :"YOU'VE GOT $";STR$(T)
```

```
260 IF T<=0 THEN 960
270 PRINT :"HOW MUCH TO DO YOU BET?"
280 INPUT "$":B
290 IF B<=T THEN 320
300 PRINT "YOU DON'T HAVE THAT MUCH"
310 GOTO 250
320 PRINT :"ENTER 8 DIFFERENT NUMBERS":"BETWEEN 1 AND 80": :
330 FOR I=1 TO 8
340 INPUT "ENTER #(1-80):":YN(I)
350 IF (YN(I)=INT(YN(I)))*(YN(I)>0)*(YN(I)<81)THEN 390
360 DISPLAY ")NUMBER MUST BE 1 TO 80(":")        TRY AGAIN       ("
370 CALL SOUND(200,110,0)
380 GOTO 340
390 IF I=1 THEN 460
400 FOR J=1 TO I-1
410 IF YN(I)<>YN(J)THEN 450
420 PRINT ")DUPLICATE NUMBER(":")    TRY AGAIN     ("
430 CALL SOUND(200,110,0)
440 GOTO 340
450 NEXT J
460 NEXT I
470 CALL CLEAR
480 PRINT "NOW THE COMPUTER WILL CHOOSE20 NUMBERS AT RANDOM"
490 FOR I=1 TO 20
500 CN(I)=INT(RND*80)+1
510 IF I=1 THEN 550
520 FOR J=1 TO I-1
530 IF CN(I)=CN(J)THEN 500
540 NEXT J
550 CALL SOUND(50,110+RND*700,3)
560 NEXT I
570 CALL CLEAR
580 PRINT "YOUR NUMBERS ARE:"
590 FOR I=1 TO 8
600 PRINT YN(I);
610 NEXT I
620 PRINT : :"COMPUTER CHOSE THESE NUMBERS"
630 FOR I=1 TO 20
640 PRINT CN(I);
650 NEXT I
660 PRINT : :"YOU MATCHED THESE NUMBERS:"
670 M=0
680 FOR I=1 TO 8
690 FOR J=1 TO 20
700 IF YN(I)<>CN(J)THEN 740
710 PRINT YN(I);
720 M=M+1
730 GOTO 750
740 NEXT J
750 NEXT I
760 IF M>0 THEN 810
770 PRINT "SORRY-NO MATCHES"
780 CALL SOUND(300,110,0)
790 T=T-B
800 GOTO 890
810 IF M=5 THEN 980
820 IF M=6 THEN 1020
830 IF M=7 THEN 1070
840 IF M=8 THEN 1130
850 PRINT : :"YOU MATCHED ONLY";M:"OUT OF 8":"SORRY-NO WINNINGS THIS TIME"
860 T=T-B
870 CALL SOUND(200,400,0)
880 CALL SOUND(200,300,0)
890 PRINT :"PLAY AGAIN? (Y/N)"
900 CALL KEY(0,KEY,STATUS)
```

```
910 Z=RND
920 IF STATUS=0 THEN 900
930 IF (KEY=89)+(KEY=121)THEN 230
940 IF (KEY=78)+(KEY=110)THEN 960
950 GOTO 900
960 PRINT "GAME OVER":"YOU END UP WITH $";STR$(T):"GOOD LUCK NEXT TIME"
970 END
980 GOSUB 1200
990 PRINT :"YOU GOT 5 NUMBERS RIGHT":"YOU WIN $";B*10;"!"
1000 T=T+B*10
1010 GOTO 890
1020 GOSUB 1200
1030 GOSUB 1200
1040 PRINT :"YOU GOT 6 NUMBERS RIGHT":"YOU WIN $";B*100;"!"
1050 T=T+B*100
1060 GOTO 890
1070 GOSUB 1200
1080 GOSUB 1240
1090 PRINT :"WOW! YOU GOT 7 NUMBERS RIGHT":"YOU WIN $";B*2000;"!!"
1100 T=T+B*2000
1110 PRINT :"GOOD WORK!"
1120 GOTO 890
1130 GOSUB 1200
1140 GOSUB 1240
1150 CALL SOUND(1,40000,30)
1160 GOSUB 1240
1170 PRINT "JACKPOT!!":"YOU GUESSED ALL 8 NUMBERS":"CORRECTLY!":"YOU WIN $";B*20
000;"!!!": :"LUCKY!"
1180 T=T+B*20000
1190 GOTO 890
1200 CALL SOUND(200,392,0)
1210 CALL SOUND(100,330,0)
1220 CALL SOUND(200,392,0)
1230 RETURN
1240 CALL SOUND(150,392,0,330,2,262,2)
1250 CALL SOUND(1000,392,0,330,2,262,2)
1260 RETURN
1270 CALL CLEAR
1280 CALL SCREEN(16)
1290 CALL CHAR(96,"0038545038145438")
1300 CALL COLOR(9,13,1)
1310 CALL HCHAR(1,1,96,768)
1320 T$="TI  KENO"
1330 FOR R=10 TO 14
1340 CALL HCHAR(R,10,32,12)
1350 NEXT R
1360 FOR C=1 TO 8
1370 CALL HCHAR(12,C+11,ASC(SEG$(T$,C,1)))
1380 NEXT C
1390 FOR I=1 TO 8
1400 CALL SOUND(50,220+20*I,0,330+20*I,0)
1410 CALL COLOR(9,1,13)
1420 FOR D=1 TO 60
1430 NEXT D
1440 CALL COLOR(9,13,1)
1450 FOR D=1 TO 60
1460 NEXT D
1470 NEXT I
1480 CALL CLEAR
1490 RETURN
```

TI-GUESS

No book of BASIC programs would be complete without a simple number guessing game. However, here is one with a twist. Instead of having the computer randomly choose a number for you to guess, this time YOU get to choose the number (an integer between 1 and 100), and the computer must guess it. You might be surprised at how well the computer does sometimes, taking as few or fewer guesses as many humans.

```
100 REM *TI-GUESS* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 RANDOMIZE
140 CALL CLEAR
150 PRINT "THINK OF A NUMBER FROM":"1 TO 100.": :"(MUST BE AN INTEGER":" )=1 AND
 (=100)"
160 PRINT :"I WON'T LOOK WHILE YOU ENTER":"YOUR NUMBER:"
170 INPUT N
180 IF (N<1)+(N)100)+(N<>INT(N))THEN 150
190 IF T>0 THEN 270
200 CALL CLEAR
210 PRINT "OKAY, NOW I'LL TRY TO GUESS":"IT.":"YOU MUST TELL ME EACH":"TIME I GU
ESS IF THE NUMBER"
220 PRINT "I GUESS IS TOO HIGH OR TOO":"LOW OR IF IT IS CORRECT.": :"IF I'M TOO
HIGH PRESS H"
230 PRINT "IF I'M TOO LOW PRESS L":"IF CORRECT PRESS C": :"NO CHEATING!": :"PRES
S ANY KEY TO BEGIN"
240 CALL KEY(0,KEY,STATUS)
250 Z=RND
260 IF STATUS=0 THEN 240
270 CALL CLEAR
280 T=0
290 A=1
300 B=100
310 R=INT((B-A+1)*RND)+A
320 PRINT "I GUESS:";R
330 T=T+1
340 PRINT : :"(ENTER H,L OR C)"
350 INPUT A$
360 IF (R=N)*(A$<>"C")THEN 470
370 IF A$<>"H" THEN 400
380 B=R-1
390 GOTO 310
400 IF A$<>"L" THEN 430
410 A=R+1
420 GOTO 310
430 IF A$="C" THEN 460
440 PRINT "INVALID ENTRY--TRY AGAIN"
450 GOTO 350
460 IF R=N THEN 510
470 PRINT "I'M GOING TO PEEK NOW.":"AHAH! THE NUMBER WAS";N:"YOU TRIED TO FOOL M
E!"
480 PRINT :"NOT NICE TO CHEAT!"
490 CALL SOUND(1000,110,0)
500 GOTO 550
510 FOR I=1 TO 10
520 CALL SOUND(50,110+50*I,0)
530 NEXT I
540 PRINT "I GUESSED IT IN";T;"TRIES! CAN YOU DO BETTER?"
550 PRINT : :"WANT TO TRY AGAIN? (Y/N)"
560 INPUT Y$
570 IF Y$="Y" THEN 140
```

MUSIC DURATION TRANSLATOR

This is a program to help you write CALL SOUND routines in your TI BASIC or TI Extended BASIC programs. One of the many outstanding features of the TI Home Computer is its ability to make music from simple commands in TI BASIC programs. However, if you have some sheet music of a tune that you wish to use in one of your programs, figuring out the duration in milliseconds for each note (as required in CALL SOUND statements) can be a very tedious process if your song is more than a couple of measures long.

The program very simply asks you to enter your tempo, then it provides you with a helpful chart showing the duration in milliseconds for each kind of note or rest from a whole all the way down to a sixteenth. The chart may be printed out for reference if you have a printer.

Here is a programming tip. When using CALL SOUND, you can use an inaudible frequency as a rest. For example, CALL SOUND (500,40000,30) tells the computer to play an inaudible tone (40,000 hz at -30db) for a half second. Also, if you want to be sure the entire duration of a tone is played before the computer executes the next program line, put an inaudible "stop note" in, such as CALL SOUND (1,40000,30).

Remember in programming long notes that 4250 milliseconds is the longest duration you can specify in 1 note, but you can use 2 or more identical notes back to back. Notes that long, though, don't make for very lively songs.

```
100 REM *MUSIC DURATION TRANSLATOR*  TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CLEAR
140 PRINT TAB(6);"* MUSIC DURATION *";TAB(6);"*   TRANSLATOR   *":::::::::
 :::::
150 INPUT "USING A PRINTER? (Y/N)":Y$
160 IF Y$<>"Y" THEN 190
170 INPUT "DEVICE NAME:":DN$
180 OPEN #1:DN$,OUTPUT
190 CALL CLEAR
200 PRINT TAB(9);"INSTRUCTIONS":::"ENTER THE TEMPO OF THE MUSIC":"IN BEATS PER M
INUTE."
210 PRINT :::"THE COMPUTER WILL GIVE THE":"DURATION OF EACH NOTE IN":"MILLISECON
DS"::
220 PRINT "THIS SHOULD FACILITATE":"WRITING MUSIC PROGRAMS.":::"(PRESS ANY KEY T
O CONTINUE)"
230 CALL KEY(0,KEY,STATUS)
240 IF STATUS=0 THEN 230
250 CALL CLEAR
260 PRINT "TAILOR THE TEMPO TO YOUR":"TASTE, BUT YOU MAY USE THESE":"AS A GUIDE:
":::
270 PRINT " TEMPO     APPROX.BEATS/MIN.":::
280 PRINT "LARGO (SLOW)";TAB(20);"42-70":"LARGHETTO";TAB(20);"70-98":"ADAGIO";TA
B(20);"98-126"
290 PRINT "ANDANTE";TAB(20);"126-154":"ALLEGRO";TAB(20);"154-182":"PRESTO (FAST)
";TAB(20);"182-210"::
```

```
300 PRINT : : :"(PRESS ANY KEY TO CONTINUE)"
310 CALL KEY(0,KEY,STATUS)
320 IF STATUS=0 THEN 310
330 CALL CLEAR
340 INPUT "TEMPO ":T
350 T$="TEMPO="&STR$(T)
360 PRINT :"ENTER": :"  1. QUARTER NOTE=1 BEAT":"         (4/4,2/4,3/4 TIME)": :
370 PRINT "  2. EIGHTH NOTE=1 BEAT":"       (6/8 TIME)": :
380 INPUT TIME
390 IF TIME=1 THEN 440
400 IF TIME<>2 THEN 380
410 B$="EIGHTH NOTE=1 BEAT"
420 T=T/2
430 GOTO 450
440 B$="QUARTER NOTE=1 BEAT"
450 CALL CLEAR
460 Q=INT(60/T*1000)
470 S=INT(.25*Q)
480 E=INT(.5*Q)
490 DE=INT(.75*Q)
500 DQ=INT(1.5*Q)
510 H=INT(2*Q)
520 DH=INT(3*Q)
530 W=INT(4*Q)
540 W$="WHOLE NOTE..... "&STR$(W)
550 DH$="DOTTED HALF.... "&STR$(DH)
560 H$="HALF NOTE...... "&STR$(H)
570 DQ$="DOTTED QUARTER. "&STR$(DQ)
580 Q$="QUARTER NOTE... "&STR$(Q)
590 DE$="DOTTED EIGHTH.. "&STR$(DE)
600 E$="EIGHTH NOTE.... "&STR$(E)
610 S$="SIXTEENTH NOTE. "&STR$(S)
620 DISPLAY T$: :B$: :W$:DH$: :H$:DQ$: :Q$:DE$: :E$:S$: :
630 IF DN$="" THEN 650
640 PRINT #1:T$: :B$: :W$:DH$: :H$:DQ$: :Q$:DE$: :E$:S$:
650 INPUT "ANOTHER ENTRY? (Y/N) ":Y$
660 IF Y$="Y" THEN 330
670 END
```

MUSIC FREQUENCY TRANSLATOR

Like the Duration Translator, this program helps you to write CALL SOUND statements and optionally gives you a printed record of conversions for future reference. You are asked to enter the name of the note (by letter) that you wish to have converted to its frequency. Use a plus sign for Sharp and a minus sign for Flat. You are also asked to specify the octave (1-4). Middle C would be in the second octave. When you are finished, simply press Enter at the "Note" prompt to end the program.

```
100 REM *MUSIC FREQUENCY TRANSLATOR* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM N(21)
140 DATA 110,123,131,147,165,175,196
150 DATA 117,131,139,156,175,185,208
160 DATA 208,117,123,139,156,165,185
170 FOR I=1 TO 21
180 READ N(I)
```

```
190 NEXT I
200 CALL CLEAR
210 DISPLAY " MUSIC FREQUENCY TRANSLATOR": :"ENTER EACH NOTE BY LETTER  AND A P
LUS (+) FOR SHARP OR A MINUS (-) FOR FLAT.": :"EXAMPLE:"
220 DISPLAY " A+ IS A-SHARP":" B- IS B-FLAT":" F  IS F-NATURAL": :"OCTAVE NUMBER
S MUST BE":"BETWEEN 1 (LOW) AND 4 (HIGH)"
230 DISPLAY :"THE COMPUTER WILL PRINT THE FREQUENCY FOR THE NOTE."
240 DISPLAY :"THE FREQUENCY RANGE IS:":" LOW A = 110 HZ":" HIGH A-FLAT = 1664 HZ
": :
250 INPUT "USING A PRINTER? (Y/N)":Y$
260 IF Y$<>"Y" THEN 290
270 INPUT "DEVICE NAME:":DN$
280 OPEN #1:DN$,OUTPUT
290 CALL CLEAR
300 DISPLAY "PRESS ENTER FOR NOTE TO QUIT": :
310 INPUT "NOTE:":N$
320 IF N$="" THEN 490
330 INPUT "OCTAVE:":OC
340 IF (OC<1)+(OC>4)+(OC<>INT(OC))+(ASC(N$)<65)+(ASC(N$)>71)+(LEN(N$)>2)THEN 310
350 IF POS(N$,"+",1)=0 THEN 390
360 F=N(ASC(N$)-57)*2^(OC-1)
370 P$=SEG$(N$,1,1)&"-SHARP, OCTAVE "&STR$(OC)
380 GOTO 450
390 IF POS(N$,"-",1)=0 THEN 430
400 F=N(ASC(N$)-50)*2^(OC-1)
410 P$=SEG$(N$,1,1)&"-FLAT, OCTAVE "&STR$(OC)
420 GOTO 450
430 F=N(ASC(N$)-64)*2^(OC-1)
440 P$=N$&", OCTAVE "&STR$(OC)
450 PRINT :F: :
460 IF DN$="" THEN 310
470 PRINT #1:P$;TAB(24);F
480 GOTO 310
490 STOP
```

CASSETTE PROGRAM FINDER

Place this short TI BASIC program at the beginning of each of your program cassettes to keep track of the location of each program on the tape. It will only take seconds to load and will provide you a handy screen menu of the titles of programs and the tape counter number of each so you can quickly fast-forward to the one you need. Optionally, you can use a printer to have a permanent catalog or to print labels for the tape boxes. The program titles listed in the Data statements are merely examples, so of course you would want to substitute your own data. Just remember to use a number sign (#) as the last piece of data so the computer will know to stop. Also, you must place the tape number (or name or letter) as the first piece of data. Leave a few seconds of tape space after the program to allow for additions.

```
100 REM *CASSETTE PROGRAM FINDER* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CLEAR
140 READ C$
150 DISPLAY "TAPE #";C$: :"  TITLE";TAB(20);"COUNTER": : :
160 READ T$
```

```
170 IF T$="#" THEN 210
180 READ N$
190 DISPLAY T$;TAB(22);N$
200 GOTO 160
210 DISPLAY : : :
220 INPUT "PRINT LABEL OR LIST? (Y/N)":Y$
230 IF Y$()"Y" THEN 340
240 INPUT "INPUT DEVICE NAME:":D$
250 OPEN #1:D$,OUTPUT
260 RESTORE
270 READ C$
280 PRINT #1:"TAPE #";C$
290 READ T$
300 IF T$="#" THEN 340
310 READ N$
320 PRINT #1:N$;TAB(5);T$
330 GOTO 290
340 STOP
350 REM TAPE NUMBER
360 DATA 5
370 REM PROG NAME,COUNTER #
380 DATA AIRLINE GUIDE,0
390 DATA DICE GAME,34
400 DATA SPRITE FUN (XB),55
410 DATA #
420 REM LAST DATA SHOULD BE NUMBER SIGN
```

AIRLINE GUIDE

Anyone that travels often and on a moment's notice knows what an inconvenience it is to call around to the various airlines to see which ones have direct flights to the cities they need to visit. Being put on hold by an airline or being told that your travel agent is busy talking on another line causes frustrating delays.

Your computer can tell you in a moment which carriers fly directly to your most frequently visited destinations, provide you with the telephone numbers for each airline that services your area, and even remind you of which companies offer special fares or especially convenient service to the city of your choosing.

The information in this TI BASIC program is contained in Data statements to facilitate updating and loading the data. The cities and airlines listed are for illustrative purposes only, so the destinations (C$) and the name of your local airport would have to be changed to suit your situation.

Since airline routes are always changing, don't forget to update the data when you get a new system timetable from one of your favorite airlines.

A printer can be used with this program, allowing you to have a handy printout of which airlines service a particular city and the appropriate phone numbers. Optionally, you can print a list of all cities in alphabetical order with the airlines that service them.

When writing your Data statements, follow these conventions. The first piece of data is the name of the airline, read as A$, followed by that company's phone number (N$). You may want to use the local number or a national 800 number in case you call from another locale. Then, the cities serviced by that carrier from your local airport are listed.

At the end of the city list for each airline, use a number sign (#) as a separate piece of data to tell the computer that the next item is going to be another airline. If an airline offers special fares or special convenient service to a particular city, you may use an asterisk (*) as a separate piece of data immediately following that city. When the computer reads this, it knows to print a message that special service is offered on this route. Remember to end each airline's city listing with a "#". At the very end of the Data statements, use a double number sign (##) twice (separated by a comma) as the last two pieces of data (*i.e.*, DATA ##,##). When read as the airline (A$) and phone number (N$), it will signal the end of the list. The array for the sort routine is initialized only when you request the complete list of cities.

```
100 REM *AIRLINE GUIDE* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CLEAR
140 S$="                                "
150 DISPLAY "WANT TO USE A PRINTER? (Y/N)"
160 INPUT Y$
170 IF Y$<>"Y" THEN 210
180 INPUT "PRINTER DEVICE NAME:":D$
190 OPEN #1:D$,OUTPUT
200 PRINT #1:TAB(8);"AIRLINE GUIDE"
210 DISPLAY "ENTER":"1 - TO SEARCH BY CITY":"2 - TO PRINT ALL CITIES"
220 INPUT YP
230 YC$=" DALLAS/FT.WORTH "
240 IF YP=2 THEN 630
250 DISPLAY :"WHAT CITY WILL YOU VISIT?"
260 INPUT W$
270 DISPLAY : :"THESE AIRLINES FLY BETWEEN":YC$;"AND":W$;":": :
280 IF Y$<>"Y" THEN 300
290 PRINT #1: :TAB(8);"THESE AIRLINES FLY BETWEEN";YC$;"AND ";W$;":":
300 RESTORE
310 M=0
320 READ A$,N$
330 IF A$="##" THEN 470
340 READ C$
350 IF C$="#" THEN 320
360 IF C$<>W$ THEN 340
370 M=M+1
380 DISPLAY :A$;TAB(LEN(A$)+3);N$
390 READ C$
400 IF C$<>"*" THEN 420
410 DISPLAY TAB(3);"(SPECIAL FARE OR SERVICE)"
420 IF Y$<>"Y" THEN 350
430 PRINT #1:TAB(8);A$;TAB(LEN(A$)+15);N$
440 IF C$<>"*" THEN 460
450 PRINT #1:TAB(5);"(SPECIAL FARE OR SERVICE)":
460 GOTO 350
470 IF M>0 THEN 490
480 DISPLAY "NO MATCH FOUND":"CHECK SPELLING": : :
```

```
490 DISPLAY : :"ANOTHER CITY? (Y/N)"
500 INPUT Z$
510 IF Z$="Y" THEN 250
520 STOP
530 DATA REPUBLIC,988-1855,ABERDEEN,BISMARK,BOSTON,BURBANK,FARGO,GRAND FORKS,HOU
STON,KANSAS CITY,LAS VEGAS
540 DATA MINOT,OMAHA,ORANGE COUNTY,SIOUX CITY,SIOUX FALLS,TORONTO,TUCSON,WATERTO
WN,#
550 DATA TWA,741-6741,NEW YORK,*,WASHINGTON,#
560 DATA DELTA,630-3200,ATLANTA,AUSTIN,BALTIMORE,BOSTON,CHICAGO,DENVER,*,HOUSTON
,JACKSON,KANSAS CITY,LAS VEGAS
570 DATA LOS ANGELES,NEW YORK,NEWARK,PHOENIX,ST.LOUIS,SAN ANTONIO,SAN FRANCISCO
580 DATA WASHINGTON,#
590 DATA AMERICAN,267-1151,AMARILLO,AUSTIN,BALTIMORE,BOSTON,CHICAGO,CORPUS CHRIS
TI,DENVER,EL PASO,HARLINGEN
600 DATA HOUSTON,JACKSON,LAS VEGAS,LOS ANGELES,LUBBOCK,MIDLAND,ODESSA,NEW YORK,N
EWARK,OKLAHOMA CITY
610 DATA ONTARIO,PHOENIX,ST.LOUIS,SAN ANTONIO,SAN FRANCISCO,SAN JOSE,TULSA,WASHI
NGTON,#
620 DATA ##,##
630 DISPLAY "SORTING-PLEASE WAIT"
640 DIM P$(200)
650 RESTORE
660 I=0
670 READ A$,N$
680 IF A$="##" THEN 780
690 READ C$
700 IF C$="#" THEN 670
710 IF C$()"*" THEN 740
720 P$(I)=P$(I)&"(SPECIAL)"
730 GOTO 690
740 C$=SEG$(C$&S$,1,28)&A$&"    "&N$
750 I=I+1
760 P$(I)=C$
770 GOTO 690
780 X=1
790 X=2*X
800 IF X(=I THEN 790
810 X=INT(X/2)
820 IF X=0 THEN 940
830 FOR J=1 TO I-X
840 Y=J
850 Z=Y+X
860 IF P$(Y)(=P$(Z)THEN 920
870 Q$=P$(Y)
880 P$(Y)=P$(Z)
890 P$(Z)=Q$
900 Y=Y-X
910 IF Y)0 THEN 850
920 NEXT J
930 GOTO 810
940 DISPLAY :"PRESS A KEY TO PAUSE": :
950 CALL SOUND(150,500,0)
960 FOR J=1 TO I
970 DISPLAY :SEG$(P$(J),1,28):SEG$(P$(J),29,28)
980 IF Y$()"Y" THEN 1000
990 PRINT #1:P$(J)
1000 CALL KEY(0,K,S)
1010 IF S=0 THEN 1030
1020 INPUT "(PRESS ENTER TO CONTINUE)":AG$
1030 NEXT J
1040 INPUT "PRINT AGAIN? (Y/N)":AG$
1050 IF AG$="Y" THEN 960
1060 STOP
```

KEYWORD ARTICLE SEARCH

When you bought your computer, you might have hoped it would cut down on the amount of paper you handle. But, if you are like most computer enthusiasts, you have found yourself in the middle of an "information explosion" and read several computer magazines regularly. If you find yourself saying, "Where was the article I read a few months ago on that subject?" and digging through stacks of magazines randomly to find that helpful tidbit of information, this TI BASIC program could help you organize your magazine collection. When you see an article you may wish to find again later, add the title of the article, the source (name of publication), date, page number, and any other key subject words you may want to use to search for the article, in that order, to the Data statements. If you wish to leave any one of these items blank in an entry, put a comma in the Data statement for that item so you will not be reprimanded by your computer for not including the correct number of items in your data list. The very last piece of data should be a number sign (#), which signals the computer to stop searching.

To find that elusive article, just enter a keyword. The computer will look through both the titles and subject words for a match. If it finds one, it prints it and asks you if it is the one you were looking for. If it is not, it keeps searching until it runs out of data and asks if you would like to try searching by another keyword. Keeping the magazines stacked in order by date or title of publication is your job.

```
100 REM *KEYWORD ARTICLE SEARCH* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CHAR(95,"00FF0000000000000")
140 CALL SCREEN(8)
150 CALL CLEAR
160 INPUT "ENTER KEYWORD:":K$
170 PRINT :"SEARCHING..."
180 M=0
190 RESTORE
200 READ T$
210 IF T$<>"#" THEN 250
220 IF M>0 THEN 370
230 PRINT :"SORRY, NO MATCH FOUND": :
240 GOTO 370
250 READ S$,D$,P$,W$
260 IF POS(T$&W$,K$,1)=0 THEN 200
270 M=M+1
280 CALL CLEAR
290 PRINT "Title:":"_____":T$: :"Source: ";S$
300 PRINT "Date: ";D$:"Page: ";P$: : :"Subject keywords:":"_____":W$:
 : :"_____"
310 PRINT :"IS THIS THE ITEM THAT YOU":"WANTED? (Y/N)"
320 CALL SOUND(100,1000,0)
330 CALL KEY(0,KEY,STATUS)
340 IF STATUS=0 THEN 330
350 IF KEY=89 THEN 370
360 IF KEY=78 THEN 200 ELSE 330
370 PRINT :"PRESS S TO SEARCH MORE":"      Q TO QUIT"
380 CALL SOUND(100,1000,0)
390 CALL KEY(0,KEY,STATUS)
```

```
400 IF STATUS=0 THEN 390
410 IF (KEY=83)+(KEY=115)THEN 150
420 IF (KEY<>81)*(KEY<>113)THEN 390
430 CALL CLEAR
440 STOP
450 REM TITLE,SOURCE,DATE,PAGE,SUBJECT WORDS
460 DATA TEACHING WITH GAMES,COMPUTE,OCTOBER 1982,98,EDUCATION
470 DATA COMPUTER GAMES OF TOMORROW,COMPUTE,OCTOBER 1982,20,VIDEODISCS
480 DATA TI MUNCHMAN,CREATIVE COMPUTING,OCTOBER 1982,65,GAMES REVIEWS
490 DATA WHAT IS LOGO?,CREATIVE COMPUTING,OCTOBER 1982,112,EDUCATION
500 DATA YOUNG PEOPLE'S LOGO ASSOCIATION,BYTE,AUGUST 1982,333,PROGRAMMING LANGUA
GES
510 DATA MINSPEAK,BYTE,SEPTEMBER 1982,186,SPEECH SYNTHESIZERS
520 DATA NOTES ON A COMPUTER SCORE,99ER MAGAZINE,VOLUME 1 NUMBER 5,68,MUSIC EDUC
ATION
530 DATA VOICE COMMAND,PERSONAL COMPUTING,NOVEMBER 1982,50,SPEECH RECOGNITION
540 DATA #
```

PERSONAL BANKING

You can keep track of your banking transactions using files created by this TI BASIC program and store them on tape, on disk, or in the TI Mini-Memory module. Set up a file for each account, then create new files periodically. The program will handle up to 198 records per file. However, keep in mind that using Extended Basic with disk and without Memory Expansion allows you less memory for data than using TI BASIC without disk. Disk and Mini-Memory both provide quick loading of data, while Extended Basic speeds program execution. Mini-Memory can only store about 125 records, so to prevent the file from over-filling, you can change line 2130 to read "If L is greater than 124 then 3150," when using that storage method.

After you have loaded a file, you may display or print the transactions, update the information by adding new transactions, or sort the data by date. The frequency at which you begin a new file depends on the number of transactions you have. You might want to start a new file every month or so. For permanent storage, you will need to save files on tape or disk, as the Mini-Memory will only hold one file at a time. (When using the Mini-Memory module, turn off the console before inserting or removing the cartridge to avoid data loss.) Your balance is automatically updated each time you add a transaction to the file.

```
100 REM *PERSONAL BANKING* TI BASIC
110 REM  BY DAVID MIGICOVSKY, BRIAN MADIGAN AND STEVE DAVIS
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 DIM A$(200)
150 DEF B$=STR$(INT(BAL))&"."&SEG$(STR$(BAL-INT(BAL))&"000",2,2)
160 CALL CLEAR
170 CALL SCREEN(16)
180 CALL CHAR(100,"0000000030408111F")
190 CALL CHAR(101,"22204F6F4F8E8EEE")
200 CALL CHAR(102,"0739D22444887F88")
210 CALL CHAR(103,"1000FCFEFE0E0E0E")
220 CALL CHAR(104,"E09C4B242211FE11")
```

```
230 CALL CHAR(105,"08007F7F7F70707F")
240 CALL CHAR(106,"000000C0201088F8")
250 CALL CHAR(107,"4404C2E6F27171F7")
260 CALL CHAR(108,"8F8F8F6E4E4E2E20")
270 CALL CHAR(109,"1F1108040300000")
280 CALL CHAR(110,"FEFEFC00F8171010")
290 CALL CHAR(111,"887F884424D23907")
300 CALL CHAR(128,"00000000000000FF")
310 CALL CHAR(129,"000000000000FF00")
320 CALL CHAR(130,"0000FF0000000000")
330 CALL CHAR(112,"7F7F70707F7F7F00")
340 CALL CHAR(113,"11FE1122244B9CE0")
350 CALL CHAR(114,"41F17176F2E2C404")
360 CALL CHAR(115,"F8881020C0000000")
370 CALL COLOR(2,6,6)
380 CALL COLOR(3,8,8)
390 CALL COLOR(4,5,5)
400 CALL COLOR(13,2,5)
410 CALL COLOR(9,5,8)
420 CALL COLOR(10,5,8)
430 CALL COLOR(11,5,8)
440 CALL COLOR(5,2,8)
450 CALL COLOR(6,2,8)
460 CALL COLOR(7,2,8)
470 CALL COLOR(8,2,8)
480 CALL HCHAR(18,4,42,6)
490 CALL HCHAR(19,4,42)
500 CALL HCHAR(19,9,42)
510 CALL HCHAR(20,4,42)
520 CALL HCHAR(20,9,42)
530 CALL HCHAR(21,4,42)
540 CALL HCHAR(21,9,42)
550 CALL HCHAR(22,4,42)
560 CALL HCHAR(22,9,42)
570 CALL HCHAR(23,4,42,6)
580 R=19
590 C=5
600 GOSUB 1000
610 PRINT : : : : : : : : : :
620 CALL HCHAR(6,4,53,25)
630 CALL HCHAR(7,4,53,25)
640 CALL HCHAR(5,4,53,25)
650 CALL HCHAR(13,4,42,6)
660 FOR I=7 TO 12
670 CALL HCHAR(I+1,10,60,19)
680 NEXT I
690 CALL HCHAR(14,4,53,25)
700 CALL HCHAR(15,4,53,25)
710 M$="PERSONAL"
720 X=5
730 Y=5
740 GOSUB 2520
750 M$="BANKING"
760 X=6
770 GOSUB 2520
780 REM PUT YOUR NAME IN THE NEXT LINE, USE "5" FOR BLANKS
790 M$="YOUR5NAME5HERE"
800 X=5
810 Y=15
820 GOSUB 2520
830 CALL HCHAR(9,16,130,8)
840 CALL HCHAR(10,11,128,17)
850 CALL HCHAR(11,11,128,17)
860 CALL HCHAR(13,17,129,8)
870 M$="PRESS5ANY5KEY5TO5CONTINUE"
```

```
880 X=19
890 Y=4
900 GOSUB 2520
910 CALL KEY(0,I,J)
920 IF J=0 THEN 910
930 CALL CLEAR
940 FOR I=1 TO 8
950 CALL COLOR(I,5,1)
960 NEXT I
970 GOTO 1170
980 R=3
990 C=21
1000 CALL HCHAR(R,C,100)
1010 CALL HCHAR(R,C+1,102)
1020 CALL HCHAR(R,C+2,104)
1030 CALL HCHAR(R,C+3,106)
1040 CALL HCHAR(R+1,C,101)
1050 CALL HCHAR(R+1,C+1,103)
1060 CALL HCHAR(R+1,C+2,105)
1070 CALL HCHAR(R+1,C+3,107)
1080 CALL HCHAR(R+2,C,108)
1090 CALL HCHAR(R+2,C+1,110)
1100 CALL HCHAR(R+2,C+2,112)
1110 CALL HCHAR(R+2,C+3,114)
1120 CALL HCHAR(R+3,C,109)
1130 CALL HCHAR(R+3,C+1,111)
1140 CALL HCHAR(R+3,C+2,113)
1150 CALL HCHAR(R+3,C+3,115)
1160 RETURN
1170 M$=" - LOAD ACCOUNT FILE"
1180 PRINT TAB(9);"FUNCTION": : : : :TAB(5);"1"&M$:TAB(9);"FROM CS1": :TAB(5);"2
"&M$:TAB(9);"FROM DSK1"
1190 PRINT :TAB(5);"3"&M$:TAB(9);"FROM MINI-MEMORY": :TAB(5);"4 - CREATE ACCOUNT
";TAB(9);"FILE": : : :
1200 GOSUB 980
1210 CALL SOUND(150,1200,0)
1220 CALL KEY(0,I,J)
1230 IF (J=0)+(I(49)+(I)52)THEN 1220
1240 ON I-48 GOTO 1250,1370,1400,1550
1250 OPEN #1:"CS1",INPUT ,INTERNAL,FIXED 192
1260 INPUT #1:NAM$,ACN$,DAT$,BAL
1270 I=1
1280 INPUT #1:A$(I),A$(I+1),A$(I+2),A$(I+3),A$(I+4),A$(I+5)
1290 FOR J=0 TO 5
1300 IF A$(I+J)="#" THEN 1340
1310 NEXT J
1320 I=I+6
1330 GOTO 1280
1340 L=I+J-1
1350 CLOSE #1
1360 GOTO 1960
1370 INPUT "FILENAME=DSK1.":M$
1380 M$="DSK1."&M$
1390 GOTO 1410
1400 M$="MINIMEM"
1410 CALL CLEAR
1420 OPEN #1:M$,INPUT ,INTERNAL,SEQUENTIAL,FIXED 32
1430 INPUT #1:NAM$
1440 INPUT #1:ACN$
1450 INPUT #1:DAT$,BAL
1460 I=1
1470 INPUT #1:A$(I)
1480 IF A$(I)="#" THEN 1510
1490 I=I+1
1500 GOTO 1470
```

```
1510 L=I-1
1520 CLOSE #1
1530 GOTO 1960
1540 REM CREATE FILE
1550 CALL CLEAR
1560 PRINT TAB(9);"CREATE FILE":TAB(9);"------ ----": : :"ENTER -": :
1570 INPUT "YOUR NAME:":NAM$
1580 INPUT "ACCOUNT NUMBER:":ACN$
1590 INPUT "TODAY'S DATE:":DAT$
1600 INPUT "ACCOUNT BALANCE: $":BAL
1610 A$(1)="#"
1620 L=0
1630 CALL CLEAR
1640 PRINT TAB(8);"SAVE FILE ON": :TAB(8);"1 - CS1": :TAB(8);"2 - DSK1": :TAB(8)
;"3 - MINIMEMORY": : : : : : :
1650 CALL SOUND(150,1200,0)
1660 CALL KEY(0,I,J)
1670 IF (J=0)+(I(49)+(I)51)THEN 1660
1680 ON I-48 GOSUB 1700,1810,1840
1690 GOTO 1960
1700 OPEN #1:"CS1",OUTPUT,INTERNAL,FIXED 192
1710 PRINT #1:NAM$,ACN$,DAT$,BAL
1720 I=1
1730 PRINT #1:A$(I),A$(I+1),A$(I+2),A$(I+3),A$(I+4),A$(I+5)
1740 FOR J=0 TO 5
1750 IF A$(I+J)="#" THEN 1790
1760 NEXT J
1770 I=I+6
1780 GOTO 1730
1790 CLOSE #1
1800 RETURN
1810 INPUT "FILENAME=DSK1.":M$
1820 M$="DSK1."&M$
1830 GOTO 1850
1840 M$="MINIMEM"
1850 OPEN #1:M$,OUTPUT,SEQUENTIAL,INTERNAL,FIXED 32
1860 PRINT #1:NAM$
1870 PRINT #1:ACN$
1880 PRINT #1:DAT$,BAL
1890 I=1
1900 PRINT #1:A$(I)
1910 IF A$(I)="#" THEN 1940
1920 I=I+1
1930 GOTO 1900
1940 CLOSE #1
1950 RETURN
1960 CALL CLEAR
1970 PRINT TAB(5);"1 - UPDATE ACCOUNT": :TAB(5);"2 - DISPLAY FILE": :TAB(5);"3 -
 PRINT FILE"
1980 PRINT :TAB(5);"4 - SAVE FILE": :TAB(5);"5 - LOAD ANOTHER FILE": :TAB(5);"6
- QUIT": : : : :
1990 GOSUB 980
2000 CALL SOUND(150,1200,0)
2010 CALL KEY(0,I,J)
2020 IF (J=0)+(I(49)+(I)54)THEN 2010
2030 ON I-48 GOTO 2060,2770,3040,1630,930,2040
2040 INPUT "ARE YOU SURE YOU WANT TO    QUIT? (Y/N)":M$
2050 IF M$="Y" THEN 2560 ELSE 1960
2060 CALL CLEAR
2070 PRINT TAB(5);"1 - ADD RECORDS TO FILE": :TAB(5);"2 - SORT FILE BY DATE": :T
AB(5);"3 - EXIT"
2080 CALL SOUND(150,1200,0)
2090 CALL KEY(0,I,J)
2100 IF (J=0)+(I(49)+(I)51)THEN 2090
2110 ON I-48 GOTO 2120,2580,1960
```

```
2120 CALL CLEAR
2130 IF L>197 THEN 3150
2140 PRINT TAB(8);"UPDATE ACCOUNT": :"LAST UPDATE:": :DAT$: :"BALANCE: $"&B$: :L
;"RECORDS IN FILE.": : :
2150 INPUT "TODAY'S DATE=":DAT$
2160 L=L+1
2170 PRINT :"MONTH OF TRANSACTION"
2180 INPUT "(1-12)?":M
2190 M$=STR$(M)
2200 IF M>9 THEN 2220
2210 M$="0"&M$
2220 PRINT :"DATE OF TRANSACTION"
2230 INPUT "(1-31)?":D
2240 D$=STR$(D)
2250 IF D>9 THEN 2270
2260 D$="0"&D$
2270 IF (M<1)+(M>12)+(D<1)+(D>31)THEN 2180
2280 M$=M$&" "&D$
2290 PRINT :"DESCRIPTION OF TRANSACTION"
2300 INPUT "(12 CHAR MAX)?":T$
2310 PRINT :"TYPE OF TRANSACTION:":"+ = DEPOSIT/CREDIT":"- = WITHDRAWAL/DEBIT"
2320 CALL SOUND(150,1200,0)
2330 CALL KEY(0,K,S)
2340 IF S=0 THEN 2330
2350 IF (K<>43)*(K<>45)THEN 2330
2360 PRINT :"AMOUNT OF TRANSACTION"
2370 INPUT "$":AM
2380 AM$=STR$(INT(AM))&"."&SEG$(STR$(AM-INT(AM))&"000",2,2)
2390 SP$=SEG$("                ",1,14-LEN(T$))
2400 A$(L)=M$&" "&SEG$(T$,1,12)&SP$&AM$&CHR$(K)
2410 IF K=43 THEN 2430
2420 AM=AM*(-1)
2430 BAL=BAL+AM
2440 PRINT :"ENTER ANOTHER TRANSACTION?  (Y/N)"
2450 CALL SOUND(150,1200,0)
2460 CALL KEY(0,X,Y)
2470 IF (Y=0)THEN 2460
2480 IF X=89 THEN 2160
2490 IF X<>78 THEN 2460
2500 A$(L+1)="#"
2510 GOTO 1960
2520 FOR I=1 TO LEN(M$)
2530 CALL HCHAR(X,Y-1+I,ASC(SEG$(M$,I,1)))
2540 NEXT I
2550 RETURN
2560 CALL CLEAR
2570 STOP
2580 CALL CLEAR
2590 PRINT "SORTING - PLEASE WAIT"
2600 R=1
2610 R=R*2
2620 IF R<=L THEN 2610
2630 R=INT(R/2)
2640 IF R=0 THEN 2760
2650 FOR I=1 TO L-R
2660 C=I
2670 X=R+C
2680 IF A$(C)<=A$(X)THEN 2740
2690 M$=A$(C)
2700 A$(C)=A$(X)
2710 A$(X)=M$
2720 C=C-R
2730 IF C>0 THEN 2670
2740 NEXT I
2750 GOTO 2630
```

```
2760 GOTO 1960
2770 CALL CLEAR
2780 PRINT TAB(6);"DISPLAY ACCOUNT"
2790 PRINT : :NAM$: :"ACCT #"&ACN$: :"LAST UPDATE:";DAT$: :"BALANCE: $"&B$
2800 IF L>0 THEN 2840
2810 PRINT : :"NO TRANSACTION RECORDS": :"PRESS A KEY"
2820 CALL KEY(0,K,S)
2830 IF S=0 THEN 2820 ELSE 1960
2840 PRINT : : :"THERE ARE";L;"TRANSACTIONS":"IN THE FILE.": :"BEGIN DISPLAY WIT
H WHICH"
2850 INPUT "TRANSACTION NUMBER?":X
2860 IF X>L THEN 2850
2870 C=0
2880 CALL CLEAR
2890 PRINT TAB(6);DAT$:TAB(6);"BALANCE: $"&B$
2900 PRINT :"DATE  DESCRIPTION   AMOUNT":"----  ----------   ------"
2910 FOR R=X TO L
2920 PRINT A$(R)
2930 C=C+1
2940 IF C<16 THEN 3000
2950 PRINT :"  PRESS A KEY TO CONTINUE"
2960 CALL KEY(0,I,J)
2970 IF J=0 THEN 2960
2980 C=0
2990 CALL CLEAR
3000 NEXT R
3010 PRINT :"  PRESS A KEY TO CONTINUE"
3020 CALL KEY(0,K,S)
3030 IF S=0 THEN 3020 ELSE 1960
3040 CALL CLEAR
3050 INPUT "PRINTER DEVICE NAME=":M$
3060 OPEN #1:M$,OUTPUT
3070 I=0
3080 PRINT #1:NAM$:ACN$:DAT$:"BALANCE: $";B$: :"DATE  DESCRIPTION    AMOUNT": :
3090 I=I+1
3100 IF A$(I)="#" THEN 3130
3110 PRINT #1:A$(I)
3120 GOTO 3090
3130 CLOSE #1
3140 GOTO 1960
3150 PRINT "FILE FULL"
3160 CALL SOUND(200,110,0)
3170 FOR R=1 TO 500
3180 NEXT R
3190 GOTO 1960
```

TRIP PLANNER

Anyone who travels by car a lot knows how bothersome it gets, sitting down with the maps and figuring out an itinerary and the time needed to complete it. This TI BASIC program is designed to aid in the task. From the start, it appeared that in order to compile all the data for such a program, memory could disappear quickly under a heap of mileage figures. It was necessary to think of a way to "cheat" a little for the sake of saving precious bytes that might be consumed by superfluous Data statements.

Take, for example, the initial program draft, which was designed to handle 30 cities. That doesn't sound like many cities until you consider that for each city you

need to know the distance to each of the other 30, making a total of 900 pieces of data. Working with a cassette machine is very slow when you have to load a large data array from a file because the maximum record length for internal-format tape files on the 99/4 is 192 characters. And, the business of packing and unpacking data makes it very easy to make a mistake that can be difficult to debug.

However, to put all the information in Data statements requires paying a double penalty in memory because the data is stored in the program listing and again in the memory locations of the variables to which it is assigned when the program is run. The only alternative was to reduce the data without sacrificing the number of cities in the program. A couple of simple (and what should have been obvious) answers presented themselves. First of all, since it is the same distance from Denver to Dallas as it is from Dallas to Denver, then why bother listing the information twice? Further, why include data that tells us the obvious, namely that it is zero miles from a city to itself? If you carefully load your array, one or two lines of program can figure all of this out. Also, since the computer takes 9 bytes to store a number, and since you have to set up a string array for the city names anyway, then why not just make the mileage figures strings in the same array and save some more bytes?

The program version listed here includes 20 major cities across the U.S. You might want to use this program to plan your next vacation. If you are a salesman that regularly travels a specific region, you could customize the program by altering the data and the size of the array. The listing here in TI BASIC takes up about 6000 bytes with the array loaded, which on a 16K TI console leaves room for additional data.

The array (C$) stores the city names and the mileage figures. At line 180, the elements with a second subscript of 0 are assigned the names, and in line 190 the computer is told that the distance between a city and itself is zero miles because the subscripts are the same. Notice in line 220 that the first value of the second subscript (J) changes each time the first (I) is read. This virtually cuts the amount of data being put into the array in half, eliminating the duplicate mileage listings between two cities. Lines 270 to 300 allow the computer to get information it needs to determine your fuel cost for the journey.

The loop at line 530 takes each city you have chosen to visit, beginning with the second one, and compares it with the one before it. Because of the way the array is loaded, the city with the smaller number must become the first subscript. For example, if you want the distance between San Francisco and Chicago, and you entered city number 18 (San Francisco) and number 3 (Chicago), the computer picks the smaller of the two numbers and makes it the first subscript, making the larger number the second. In this case, it looks up C$(3, 18) and finds a string, "2155," which it converts to a number, and then informs you that it is 2155 miles from San Francisco to Chicago. Lines 660 and 670 let you know how long the trip

will take (assuming you average 55 mph with bathroom stops), and how much you will spend in gasoline money. It might be interesting to see how close you come to the cost estimated by the computer. Bon voyage!

```
100 REM *TRIP PLANNER* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM  COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM C$(20,20)
140 DEF T=5-LEN(STR$(I))
150 CALL CLEAR
160 PRINT "INITIALIZING DATA--STAND BY"
170 FOR I=1 TO 20
180 READ C$(I,0)
190 C$(I,I)="0"
200 NEXT I
210 FOR I=1 TO 19
220 FOR J=I+1 TO 20
230 READ C$(I,J)
240 NEXT J
250 NEXT I
260 CALL CLEAR
270 PRINT "HOW MANY MILES PER GALLON":"DOES YOUR CAR GET ON THE":"HIGHWAY?"
280 INPUT MPG
290 PRINT :"HOW MUCH DO YOU PAY FOR A":"GALLON OF GAS?"
300 INPUT "$":P
310 CALL CLEAR
320 FOR I=1 TO 20
330 PRINT TAB(T);I;C$(I,0)
340 NEXT I
350 PRINT "HOW MANY CITIES WILL YOU":"VISIT? (INCLUDE START &":"DESTINATION. 10
MAX.)"
360 INPUT N
370 IF N<2 THEN 360
380 IF N>10 THEN 360
390 CALL CLEAR
400 PRINT "INSTRUCTIONS:":"ENTER THE NUMBER FOR EACH":"CITY YOU WILL VISIT IN TH
E":"ORDER YOU WILL VISIT THEM."
410 PRINT "(INCLUDE YOUR START &":"DESTINATION)": : :"PRESS ANY KEY TO CONTINUE"
420 CALL KEY(0,KEY,STATUS)
430 IF STATUS=0 THEN 420
440 CALL CLEAR
450 FOR I=1 TO 20
460 PRINT TAB(T);I;C$(I,0)
470 NEXT I
480 PRINT "ENTER CITY NUMBERS IN ORDER:"
490 FOR I=1 TO N
500 INPUT A(I)
510 NEXT I
520 CALL CLEAR
530 FOR I=2 TO N
540 IF A(I)>A(I-1)THEN 570
550 M=M+VAL(C$(A(I),A(I-1)))
560 GOTO 580
570 M=M+VAL(C$(A(I-1),A(I)))
580 NEXT I
590 PRINT "YOUR TRIP FROM ";C$(A(1),0);"TO ";C$(A(N),0)
600 IF N=2 THEN 650
610 PRINT "VIA"
620 FOR I=2 TO N-1
630 PRINT C$(A(I),0)
640 NEXT I
650 PRINT "TOTALS";M;"MILES."
660 PRINT :"IT SHOULD TAKE YOU ABOUT";INT(10*M/55+.5)/10;"HOURS TO DRIVE."
```

```
670 PRINT :"YOUR FUEL COST WILL BE":"ABOUT $";INT(M/MPG*P*100)/100
680 PRINT : :"WANT TO PLAN ANOTHER TRIP?":"(Y/N)"
690 INPUT Y$
700 M=0
710 IF Y$="Y" THEN 310
720 END
730 DATA ATLANTA,BOSTON,CHICAGO,DALLAS,DENVER,DETROIT,HOUSTON,KANSAS CITY,LOS AN
GELES,MEMPHIS
740 DATA MIAMI,MINNEAPOLIS,NEW ORLEANS,NEW YORK,PHILADELPHIA,PHOENIX,ST.LOUIS,SA
N FRANCISCO,SEATTLE,WASH. D.C.
750 DATA 1065,675,805,1385,715,810,785,2205,380,665,1070,480,850,770,1815,540,25
25,2725,605
760 DATA 965,1780,1960,710,1845,1395,2985,1310,1515,1360,1535,215,305,2605,1160,
3115,3000,445
770 DATA 920,1015,270,1080,540,2050,545,1335,395,925,790,745,1735,295,2155,2090,
695
780 DATA 780,1180,240,500,1365,460,1310,955,510,1565,1485,1030,630,1760,2105,134
5
790 DATA 1265,1020,600,1045,1045,2045,790,1290,1785,1720,835,845,1275,1315,1650
800 DATA 1270,740,2340,715,1370,665,1065,620,570,1960,520,2420,2285,520
810 DATA 760,1540,575,1205,1185,355,1630,1550,1150,800,1920,2275,1410
820 DATA 1615,450,1440,455,840,1185,1110,1225,245,1815,1860,1045
830 DATA 1815,2745,1835,1890,2875,2700,390,1860,390,1155,2630
840 DATA 1000,845,390,1095,1015,1435,275,2145,2400,875
850 DATA 1730,860,1300,1220,2355,1200,3140,3400,1080
860 DATA 1240,1185,1140,1625,560,1985,1620,1190
870 DATA 1320,1240,1505,680,2280,2615,1085
880 DATA 90,2395,950,2940,2825,230,2320,875,2895,2780,140
890 DATA 1510,770,1470,2250,1060,2160,805,825,2865,2730
```

VIDEO TAPE FINDER

Can't keep up with all those movies and programs that you have recorded on video tape? This TI BASIC program stores your program titles, the tape number (or letter), and the counter location of each program in Data statements, making it easy to update the information. Put data in the order indicated: title, tape number, footage number. The last item of data should be the word "END". You may scan the entire list, if you can't decide what to watch tonight. Or, you can search for an individual title. The larger your video tape collection, the more helpful this program will be.

```
100 REM *VIDEO TAPE SEARCH* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CLEAR
140 PRINT TAB(6);"VIDEO TAPE SEARCH": : : : : : : : : : :
150 FOR DELAY=1 TO 250
160 NEXT DELAY
170 CALL CLEAR
180 RESTORE
190 INPUT "SCAN ALL TITLES? (Y/N)":Y$
200 IF Y$="Y" THEN 240
210 PRINT :"ENTER TITLE:": :
220 INPUT X$
230 GOTO 300
240 PRINT :"SCAN- 1. FAST":TAB(7);"2. SLOW"
250 N=0
260 INPUT SPD
```

```
270 PRINT :"(HOLD DOWN A KEY TO STOP)": :
280 IF SPD=1 THEN 300
290 N=250
300 FOR I=1 TO 300
310 READ M$
320 IF M$="END" THEN 440
330 READ T$,C$
340 IF Y$()"Y" THEN 410
350 PRINT :M$:" TAPE #-";T$;TAB(25);C$
360 CALL KEY(0,KEY,STATUS)
370 IF STATUS()0 THEN 440
380 FOR DELAY=1 TO 2+N
390 NEXT DELAY
400 GOTO 430
410 IF POS(M$,X$,1)=0 THEN 430
420 PRINT : :" TAPE#- ";T$;TAB(14);"COUNTER#-";C$: :M$: : :
430 NEXT I
440 PRINT : : :
450 INPUT "SEARCH AGAIN? (Y/N) ":AG$
460 IF AG$="Y" THEN 170
470 STOP
480 REM MOVIE TITLE=M$,TAPE#=T$,FOOTAGE COUNTER#=C$
490 DATA SINGIN' IN THE RAIN,A,0
500 DATA THE FOUNTAINHEAD,A,496
510 DATA THE BAD AND THE BEAUTIFUL,B,0
520 DATA SUNSET BOULEVARD,B,511
530 DATA DOCTOR ZHIVAGO PART 1,C,0
540 DATA DOCTOR ZHIVAGO PART 2,D,0
550 DATA THE 5000 FINGERS OF DOCTOR T,D,300
560 DATA A SALUTE TO HITCHCOCK,E,10
570 DATA JAZZ CONCERT,E,400
580 REM LAST ITEM SHOULD BE "END"
590 DATA END
```

COLOR BAR GRAPHS

A very helpful tool for businessmen and educators is the bar graph. With it, relative differences between similar elements can be transformed from a jumble of numbers into a clear visual representation of a point. Using various colors to help differentiate the individual elements on the chart further enhances the clarity and aesthetics of any presentation.

This short program in TI Extended BASIC is very simple to use. You may use from 2 to 4 bars on each graph and each bar may be a different color. You are asked the maximum possible value of each bar. In other words, what is 100% performance? If the goal this year for the Acme Computer Company is to have each of three representatives produce 10,000 units, then maximum performance for each representative would be 10,000. Minimum performance, of course, would be zero.

The value of each bar is the relative value of each in regard to the maximum goal. In the example mentioned, producing 7500 units would give a representative 75% performance, so his bar would extend ¾ way across the screen. The title of the graph will appear at the top of the screen, and the title caption for each bar appears directly above each bar. The maximum and minimum values appear at the lower corners of the screen.

If you are doing an audio-visual presentation and need some color bar graphs in a hurry, this program could be a big help. By photographing the screen of your monitor with a single-lens reflex camera and slide film, you could use the graphs in your slide shows. Or, by sending the video signal from your computer to a video recorder, you could tape the images for incorporation into a video presentation.

```
100 REM *COLOR BAR GRAPHS*  TI EXTENDED BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CLEAR
140 INPUT "HOW MANY BARS? (2-4):":B :: IF B<2 OR B>4 THEN 140
150 PRINT "TITLE OF GRAPH:":"(28 CHAR.MAX.)" :: INPUT T$ :: IF LEN(T$)>28 THEN 1
50
160 PRINT "MAX.POSSIBLE VALUE OF BARS  (100%):" :: INPUT MV :: IF MV<=0 THEN 160
170 FOR I=1 TO B
180 PRINT "TITLE OF BAR#";I;":":"(28 CHAR.MAX.)" :: INPUT TB$(I):: IF LEN(TB$(I)
)>28 THEN 180
190 PRINT " 2-BLACK        3-MED GREEN":" 4-LT GREEN    5-DK BLUE":" 6-LT BLUE
  7-DK RED"
200 PRINT " 8-CYAN         9-MED RED":"10-LT RED      11-DK YELLOW":"12-LT YELLOW
 13-DK GREEN":"14-MAGENTA    15-GRAY"
210 PRINT "ENTER COLOR OF BAR #";I;":" :: INPUT C(I):: IF C(I)<2 OR C(I)>15 THEN
 210
220 PRINT "ENTER VALUE OF BAR #";I;":" :: INPUT V(I):: IF V(I)<=0 OR V(I)>MV THE
N 220
230 REP(I)=32*(V(I)/MV):: IF REP(I)<1 THEN REP(I)=1
240 NEXT I
250 CALL CLEAR :: CALL SCREEN(16)
260 P$="FFFFFFFFFFFFFFFF" :: CC=96
270 FOR I=1 TO B :: CALL CHAR(CC,P$):: CALL COLOR(I+8,C(I),1):: CC=CC+8 :: NEXT
I
280 DISPLAY AT(2,15-LEN(T$)/2):T$
290 FOR I=1 TO B :: DISPLAY AT(5*I,1):TB$(I):: NEXT I
300 CC=96
310 FOR I=1 TO B :: CALL HCHAR(5*I+1,1,CC,REP(I)):: CALL HCHAR(5*I+2,1,CC,REP(I)
):: CC=CC+8 :: NEXT I
320 DISPLAY AT(24,1):"0" :: DISPLAY AT(24,28-LEN(STR$(MV))):MV
330 CALL KEY(0,KEY,STATUS)
340 IF STATUS=0 THEN 330
350 PRINT "ANOTHER GRAPH? (Y/N)" :: INPUT Y$ :: IF Y$="Y" OR Y$="y" THEN 130
360 STOP
```

BAR GRAPH PRINTER

Whereas the Color Bar Graph program displays the graphs on the screen, this version is designed to run your graphs to an 80 column printer. Of course, you may need to change the device name for the type of printer you will use, but the program should work with little or no modification with most common printers. You may enter as many bars as you wish for your graph.

```
100 REM *BAR GRAPH PRINTER* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM V(20),REP(20),TB$(20)
140 INPUT "PRINTER DEVICE NAME:":DN$
150 OPEN #1:DN$,OUTPUT
```

```
160 INPUT "HOW MANY BARS? (1-20):":B
170 IF (B(1)+(B)20)THEN 160
180 PRINT "TITLE OF GRAPH:":"(80 CHAR.MAX.)"
190 INPUT T$
200 IF LEN(T$))80 THEN 180
210 PRINT "MAX. POSSIBLE VALUE OF BARS  (100%):"
220 INPUT MV
230 IF MV(=0 THEN 210
240 FOR I=1 TO B
250 PRINT "TITLE OF BAR #";I;":":"(72 CHAR.MAX.)"
260 INPUT TB$(I)
270 IF LEN(TB$(I)))72 THEN 250
280 PRINT "ENTER VALUE OF BAR #";I;":"
290 INPUT V(I)
300 IF (V(I)(=0)+(V(I))MV)THEN 280
310 REP(I)=80*(V(I)/MV)
320 IF REP(I))=1 THEN 340
330 REP(I)=1
340 NEXT I
350 PRINT #1:TAB(41-LEN(T$)/2);T$: : : :
360 PRINT #1:"0  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  . ";MV: :
370 FOR I=1 TO B
380 A$=""
390 FOR J=1 TO REP(I)
400 A$=A$&"X"
410 NEXT J
420 PRINT #1:A$:A$:A$: :TAB(41-(LEN(TB$(I)&STR$(V(I)))+3)/2);TB$(I);" = ";STR$(V
(I)): : :
430 NEXT I
440 PRINT "ANOTHER GRAPH? (Y/N)"
450 INPUT Y$
460 IF (Y$="Y")+(Y$="y")THEN 160
470 STOP
```

USER DIRECTORY

Meeting people with similar interests is one of the most exciting aspects of using computerized data services, such as CompuServe or The Source. However, after a while, you may have met so many new folks that it is hard to remember their names and ID numbers (which are used if you want to address electronic mail or page them for an online "chat"). This program will help you keep up with this information and will print a list sorted by name or by ID/account number. The names and numbers are stored in Data statements so that they can be easily appended. In the Data statements the name is listed first, followed by the account number. The last two pieces of data must be number signs (#,#). If you do not have a printer, you may search the list by ID number or name.

```
100 REM *USER DIRECTORY* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 OPTION BASE 1
140 DIM A$(100),B$(2)
150 SP$="..................."
160 CALL CLEAR
170 PRINT "1 - SEARCH":"2 - SORT & PRINT LIST":"3 - QUIT"
180 INPUT C
```

```
190 IF C=3 THEN 840
200 IF C=2 THEN 420
210 IF C<>1 THEN 180
220 CALL CLEAR
230 RESTORE 850
240 PRINT :"1 - SEARCH BY NAME":"2 - SEARCH BY ACCT NUMBER"
250 INPUT C
260 IF (C<1)+(C>2)THEN 250
270 IF C=2 THEN 300
280 INPUT "NAME=":S$
290 GOTO 310
300 INPUT "ACCT #=":S$
310 READ B$(1),B$(2)
320 IF B$(1)="#" THEN 400
330 IF S$<>B$(C)THEN 310
340 CALL CLEAR
350 PRINT "NAME:  ";B$(1): :"ACCT#: ";B$(2)
360 PRINT : :" (PRESS A KEY TO CONTINUE)"
370 CALL KEY(0,K,S)
380 IF S=0 THEN 370
390 GOTO 160
400 PRINT :"NO MATCH FOUND"
410 GOTO 360
420 CALL CLEAR
430 RESTORE 850
440 PRINT "1 - SORT BY NAME":"2 - SORT BY ACCOUNT NUMBER"
450 INPUT C
460 PRINT :"SORTING..."
470 IF C=2 THEN 530
480 FOR I=1 TO 100
490 READ C$,D$
500 IF C$="#" THEN 580
510 A$(I)=SEG$(C$&SP$,1,20)&D$
520 NEXT I
530 FOR I=1 TO 100
540 READ C$,D$
550 IF C$="#" THEN 580
560 A$(I)=SEG$(D$&SP$,1,20)&C$
570 NEXT I
580 N=I-1
590 B=1
600 B=2*B
610 IF B<=N THEN 600
620 B=INT(B/2)
630 IF B=0 THEN 750
640 FOR I=1 TO N-B
650 C=I
660 D=C+B
670 IF A$(C)<=A$(D)THEN 730
680 AA$=A$(C)
690 A$(C)=A$(D)
700 A$(D)=AA$
710 C=C-B
720 IF C>0 THEN 660
730 NEXT I
740 GOTO 620
750 CALL CLEAR
760 INPUT "PRINTER DEVICE NAME:":DN$
770 INPUT "PRESS ENTER TO PRINT":Y$
780 OPEN #1:DN$,OUTPUT
790 FOR I=1 TO N
800 PRINT #1:A$(I)
810 NEXT I
820 CLOSE #1
```

```
830 GOTO 160
840 STOP
850 DATA JOHN SMITH,TI9431
860 DATA BILL JONES,TI9567
870 DATA MARY WHITE,TI9845
880 DATA JIM DAVIS,TI9231
890 DATA #,#
900 REM LAST TWO PIECES OF DATA MUST BE NUMBER SIGNS
```

ELECTRONIC SCRATCHPAD

How often do you write notes on slips of paper or backs of envelopes? And, how often do these scrawled tidbits of valuable information become lost forever in a heap of similar notes? You can keep all your jottings in one place and search for information by any keyword that you can remember about the notes you need with this TI Extended BASIC program. Before using the main program, you should set up a disk file called "SCRATCH" (or any other name you choose). There is a short program after the main listing that does this. When you load the file with the main program, you have the options of searching, changing or adding records. To delete a record, simply replace it with a new record using the change option. Limit the length of each record to 125 characters. It is most convenient to maintain your program and file on the same disk.

```
100 REM *ELECTRONIC SCRATCHPAD* TI EXTENDED BASIC, DISK
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM A$(200)
140 CALL CLEAR :: I=1 :: CH=0 :: PRINT "LOADING FILE"
150 OPEN #1:"DSK1.SCRATCH",INPUT ,VARIABLE 126,SEQUENTIAL,INTERNAL
160 INPUT #1:A$(I)
170 IF EOF(1)THEN 180 ELSE I=I+1 :: GOTO 160
180 CLOSE #1 :: CALL CLEAR :: PRINT I;"RECORDS IN FILE"
190 PRINT :"1=SEARCH BY KEYWORD":"2=ADD DATA":"3=END"
200 INPUT C :: IF C<1 OR C>3 THEN 200 :: ON C GOTO 210,270,330
210 INPUT "KEYWORD=":K$
220 FOR J=1 TO I :: IF POS(A$(J),K$,1)=0 THEN 260 ELSE CALL CLEAR
230 PRINT :A$(J): : :"PRESS":" S TO SEARCH MORE":" C TO CHANGE THIS RECORD":" K
FOR ANOTHER KEYWORD":" Q TO QUIT SEARCH"
240 CALL KEY(0,K,S):: IF S=0 THEN 240 ELSE IF K=83 THEN 260 ELSE IF K=75 THEN 21
0 ELSE IF K=81 THEN 190 ELSE IF K=67 THEN 250 ELSE 240
250 PRINT "RE-ENTER DATA:" :: LINPUT A$(J):: CH=CH+1 :: GOTO 230
260 NEXT J :: GOTO 190
270 CALL CLEAR
280 PRINT "ENTER # TO END.": :"ENTER DATA:"
290 LINPUT AD$ :: IF AD$="#" THEN 320
300 IF LEN(AD$)>125 THEN DISPLAY :"LINE TOO LONG--TRUNCATED!" :: CALL SOUND(100,
110,0):: AD$=SEG$(AD$,1,125):: DISPLAY :AD$
310 I=I+1 :: A$(I)=AD$ :: CH=CH+1 :: GOTO 290
320 GOTO 190
330 IF CH=0 THEN 370
340 PRINT "UPDATING FILE"
350 OPEN #1:"DSK1.SCRATCH",OUTPUT,VARIABLE 126,SEQUENTIAL,INTERNAL
360 FOR J=1 TO I :: PRINT #1:A$(J):: NEXT J :: CLOSE #1
370 STOP
```

```
100 REM *CREATE SCRATCHPAD FILE*
110 REM TO CREATE A NEW FILE, CHANGE FILENAME IN LINE 120 BELOW & 150 IN MAIN PR
OGRAM
120 OPEN #1:"DSK1.SCRATCH",OUTPUT,VARIABLE 126,SEQUENTIAL,INTERNAL
130 PRINT #1:"SCRATCHPAD"
140 CLOSE #1
```

MAIL-WRITER

Here is a great aid for those using their computers to access data services using the TI Terminal Emulator II. This TI Extended BASIC program allows you to prepare text files and saves them on disk in a format that can be used by the Auto-Logon feature of TE2. Thus, you can compose your electronic mail, bulletins and files off-line before signing on to the data service. The program works equally well with The Source and CompuServe as well as many of the local electronic bulletin board message services.

The menu screen gives you 6 options:

1 — Enter
2 — Edit
3 — Save 28 Col
4 — Save 40 Col
5 — Load 28 Col
6 — Stop

To create a file, select "Enter." This function erases any previously entered text, then it allows you to begin entering text at Page 1, Line 1. You may enter up to 9 pages with 20 lines per page. At the end of a line (28 characters maximum), press Enter to go to the next line. As each line is being entered, you may use the Insert, Delete and Erase functions of the computer to edit. When you are finished entering text, enter an equal sign ("=") at the beginning of a new line. Once you leave the Enter mode, you cannot return without erasing the text you have created.

Select the Edit mode if you want to review or change the text. In this mode, a non-flashing cursor appears to the left of the first line of text. When the cursor is displayed this way, use these key commands:

UP Arrow	— Moves cursor up 1 line
DOWN Arrow	— Moves cursor down 1 line
PROC'D	— Moves to top of next page
BACK	— Moves to top of previous page
BEGIN	— Return to menu
SPACE Bar	— Moves cursor onto line for editing

When finished editing, press BEGIN to return to the menu.

If you will ever need to re-load the text for future modification, or if you want the text to be transmitted exactly as it appeared on the screen in this program, select the SAVE 28 COL option. When a 28 column file is transmitted, it will be centered on the 40 column TE2 display.

Choosing the SAVE 40 COL option makes optimum use of the 40 column TE2 display, but this type of file cannot be re-loaded into this program for future editing. For best results, follow two conventions when entering text in the Enter or Edit modes. First, if you want to join the characters at the end of one line with those at the beginning of the next line, place a hyphen ("-") at the end of the first line. This program will automatically join the words for you. Lines not using the hyphen as the last character will be separated by a space. For instance:

```
. . . exam-
ple . . .        Becomes    example

. . . example
line . . .       Becomes    example line
```

The second convention requires that a blank line be inserted in the text if the text on the next line is to begin a new line:

```
Name
Address          Becomes    Name Address


Name                        Name
                 Becomes
Address                     Address
```

Blank lines should be used, then, when the next line will start a new paragraph. These conventions are not used in the 28 column files, which are transmitted as entered. When saving a file, use a file name of 7 characters or less.

To send your message, sign on to the service (Compuserve, Source, etc.) with TE2. When you reach a point where you are prompted to begin entering text (as in Email, SourceMail, Post, etc.), then follow these steps:

1. Place disk with text file in drive 1.
2. EXIT TE2 by pressing Control 0, which will take you to the TI title screen.
3. Select option 2 of the TE2.
4. Change the name LOGON to the name of your text file on disk.
5. Press Enter.

The text will be loaded from disk and transmitted automatically, each line appearing on your screen (transmission may be halted by pressing Control 3/Cancel). Transmission is complete when the cursor stops moving and a "beep" is heard. At that time, after pressing Enter, you may enter any additional text or system commands. Note that, upon occasion, text lines echoed back to your screen may be missing a few characters. This occurs if the echoed transmission takes place while the disk is being accessed. The text actually received by the *host* should not be affected. Remember to use the Control 0 to Exit TE2 after signing off (rather than "Quit") to make sure that any disk files you have opened are properly closed.

```
100 REM *MAIL-WRITER* TI EXTENDED BASIC
110 REM BY JOHN CLULOW AND BERNIE ELSNER
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 CALL CLEAR :: CALL SCREEN(6):: FOR I=0 TO 14 :: CALL COLOR(I,16,6):: NEXT I
150 OPTION BASE 1 :: DIM L$(9,20):: X$=RPT$("-",28)
160 DISPLAY AT(2,10)ERASE ALL:"MAIL-WRITER" :: DISPLAY AT(4,1):"OFF-LINE PREPARA
TION OF TEXT" :: DISPLAY AT(5,3):"FOR TEII AUTO LOGON FILE"
170 DISPLAY AT(6,1):X$ :: DISPLAY AT(8,8):"1  ENTER" :: DISPLAY AT(10,8):"2  EDI
T"
180 DISPLAY AT(12,8):"3  SAVE 28 COL" :: DISPLAY AT(14,8):"4  SAVE 40 COL"
190 DISPLAY AT(16,8):"5  LOAD 28 COL" :: DISPLAY AT(18,8):"6  STOP" :: DISPLAY A
T(22,5):"YOUR CHOICE (1-6)? 1"
200 ACCEPT AT(22,24)VALIDATE("123456")SIZE(-1)BEEP:X
210 ON X GOTO 220,220,400,400,510,720
220 P=1 :: IF X=2 THEN F=1 ELSE F=0
230 IF F=1 THEN 250
240 FOR I=1 TO 9 :: FOR J=1 TO 20 :: L$(I,J)="" :: NEXT J :: NEXT I
250 DISPLAY AT(1,12)ERASE ALL:"PAGE";P :: DISPLAY AT(2,1):X$ :: DISPLAY AT(23,1)
:X$
260 IF F=0 THEN DISPLAY AT(24,7):"ENTER ""="" TO STOP" ELSE DISPLAY AT(24,3):"PR
OC'D    BACK    BEGIN"
270 IF F=0 THEN 290
280 FOR L=1 TO 20 :: DISPLAY AT(L+2,1):L$(P,L):: NEXT L
290 L=3
300 IF L<3 THEN L=3
310 IF F=0 THEN 380
320 CALL HCHAR(L,2,30)
330 CALL KEY(0,K,S):: IF S=0 THEN 330
340 IF K>31 OR K=-1 THEN CALL HCHAR(L,2,32):: GOTO 380 ELSE IF K=11 THEN CALL HC
HAR(L,2,32):: L=L-1 :: GOTO 300
350 IF K=14 THEN 160 ELSE IF K=15 THEN P=P-1 ELSE IF K=12 THEN P=P+1 ELSE CALL H
CHAR(L,2,32):: GOTO 390
360 IF P<1 THEN P=1 ELSE IF P>9 THEN P=9
370 GOTO 250
380 ACCEPT AT(L,1)SIZE(-28):Y$ :: L$(P,L-2)=Y$ :: IF F=0 AND Y$="=" THEN L$(P,L-
2)="" :: GOTO 160
390 L=L+1 :: IF L>22 THEN P=P+1 :: L=1 :: GOTO 360 ELSE GOTO 300
400 REM SAVE 28 COL TEXT FILE
410 IF X=3 THEN Y$="28" ELSE Y$="40"
420 DISPLAY AT(5,5)ERASE ALL:"SAVE "&Y$&" COL TEXT FILE" :: DISPLAY AT(10,8):"DS
K1." :: ACCEPT AT(10,13)SIZE(7)VALIDATE(UALPHA,DIGIT)BEEP:Y$
430 FOR I=1 TO LEN(Y$):: IF SEG$(Y$,I,1)=" " THEN 420
440 NEXT I :: DISPLAY AT(13,8):"PROCEED (Y/N)? Y" :: ACCEPT AT(13,23)VALIDATE("Y
N")SIZE(-1)BEEP:R$ :: IF R$="N" THEN 160
450 DISPLAY AT(18,12):"WAIT..." :: FOR I=9 TO 1 STEP -1 :: FOR J=20 TO 1 STEP -1
 :: IF L$(I,J)<>"" THEN 470
460 NEXT J :: NEXT I :: DISPLAY AT(12,2)ERASE ALL BEEP:"WARNING -- NO TEXT PRESE
NT!" :: FOR I=1 TO 800 :: NEXT I :: GOTO 160
470 IF X=4 THEN 590
480 FIRST$="1"&RPT$(" ",6):: OPEN #1:"DSK1."&Y$
490 FOR P=1 TO I :: IF P=I THEN H=J ELSE H=20
500 FOR L=1 TO H :: PRINT #1:FIRST$&L$(P,L)&CHR$(13):: PRINT #1:"2"&CHR$(60):: N
EXT L :: NEXT P :: CLOSE #1 :: GOTO 160
510 REM LOAD 28 COL TEXT FILE
520 DISPLAY AT(5,8)ERASE ALL:"LOAD TEXT FILE" :: DISPLAY AT(10,8):"DSK1." :: ACC
EPT AT(10,13)SIZE(8)VALIDATE(UALPHA,DIGIT)BEEP:Y$
530 FOR I=1 TO LEN(Y$):: IF SEG$(Y$,I,1)=" " THEN 520
540 NEXT I :: DISPLAY AT(13,8):"PROCEED (Y/N)? Y" :: ACCEPT AT(13,23)VALIDATE("Y
N")SIZE(-1)BEEP:R$ :: IF R$="N" THEN 160
550 OPEN #1:"DSK1."&Y$
560 DISPLAY AT(18,12):"WAIT..." :: FOR P=1 TO 9 :: FOR L=1 TO 20 :: IF EOF(1)<>0
 THEN 580 ELSE LINPUT #1:Y$ :: LINPUT #1:R$
570 L$(P,L)=SEG$(Y$,8,LEN(Y$)-8):: NEXT L :: NEXT P
580 CLOSE #1 :: GOTO 160
```

```
590 REM SAVE 40 COL TEXT FILE
600 OPEN #1:"DSK1."&Y$ :: R$="" :: Y$="" :: P=1 :: L=1
610 IF P*20+L)I*20+J THEN 710
620 IF L$(P,L)="" THEN 690 ELSE IF R$="" THEN 640
630 IF SEG$(R$,LEN(R$),1)="-" THEN R$=SEG$(R$,1,LEN(R$)-1)ELSE R$=R$&" "
640 R$=R$&L$(P,L):: IF LEN(R$)<41 THEN IF L<20 THEN L=L+1 :: GOTO 610 ELSE L=1 :
: P=P+1 :: GOTO 610
650 FOR K=41 TO 1 STEP -1 :: IF SEG$(R$,K,1)=" " THEN 680
660 NEXT K :: DISPLAY AT(12,1)ERASE ALL:"ERROR - TEXT STRING > 40 CH" :: DISPLAY
 AT(14,6):"PAGE";P :: DISLAY AT(15,6):"LINE";L
670 FOR I=1 TO 800 :: NEXT I :: GOTO 160
680 Y$=SEG$(R$,K+1,LEN(R$)-K):: PRINT #1:"1"&SEG$(R$,1,K-1)&CHR$(13):: PRINT #1:
"2"&CHR$(80):: R$=Y$ :: GOTO 700
690 PRINT #1:"1"&R$&CHR$(13):: PRINT #1:"2"&CHR$(60):: PRINT #1:"1  "&CHR$(13)::
 PRINT #1:"2"&CHR$(60):: R$=""
700 IF L<20 THEN L=L+1 :: GOTO 610 ELSE L=1 :: P=P+1 :: GOTO 610
710 PRINT #1:"1"&R$&CHR$(13):: PRINT #1:"2"&CHR$(60):: CLOSE #1 :: GOTO 160
720 CALL CLEAR :: END
```

TALKING TELETYPE

You can communicate with other TI Home Computer users with this TI BASIC program that simulates the TI-CHAT feature on the SOURCE. You will need to have the computers connected via the RS232 interface, either directly or using phone modems. Your Speech Synthesizer and Terminal Emulator II module will allow you to hear the messages sent to you. Optionally you may use a printer to have a permanent copy of your conversation. If you wish to change the voice pitch that your messages are spoken in at the other terminal, you may enter "//PP SSS" at the beginning of a new line (with no spaces before it and no text after it), where PP is a pitch code number and SSS is the slope setting. (Consult the TE2 manual for a description of these settings.)

When you call the person with whom you want to communicate, the two of you must agree which person will be User #1 and which will be User #2. Then you establish a connection between your computers and both run the program. Answer the prompts at the beginning of the program. User #1 will receive a prompt to enter a message, then each user will alternately receive a prompt to enter text. If you wish to send no reply, just press Enter. Normally, when using phone modems you will want to have one unit set in the originate mode and the other in answer mode using full duplex.

```
100 REM *TALKING TELETYPE* TI BASIC
110 REM TERMINAL EMULATOR 2, SPEECH SYNTHESIZER AND RS232
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 CALL CLEAR
150 CALL SCREEN(8)
160 INPUT "ARE YOU USER 1 OR 2?":U
170 IF (U<>1)*(U<>2)THEN 160
180 PRINT
190 INPUT "DISPLAY MESSAGES RECEIVED   ON SCREEN? (Y/N)":Y$
200 PRINT
```

```
210 INPUT "USING A PRINTER? (Y/N)":YP$
220 IF YP$<>"Y" THEN 250
230 INPUT "DEVICE NAME:":DN$
240 OPEN #3:DN$,OUTPUT
250 OPEN #1:"RS232.EC.LF",VARIABLE 112
260 OPEN #2:"SPEECH",OUTPUT
270 CALL CLEAR
280 IF U=2 THEN 420
290 REM  USER 1 - DO NOT USE COMMAS IN INPUT
300 INPUT A$
310 PRINT #1:A$
320 IF YP$<>"Y" THEN 340
330 PRINT #3:A$
340 INPUT #1:B$
350 PRINT #2:B$
360 IF Y$<>"Y" THEN 380
370 DISPLAY "->"&B$
380 IF YP$<>"Y" THEN 300
390 PRINT #3:">"&B$
400 GOTO 300
410 REM USER 2
420 INPUT #1:A$
430 PRINT #2:A$
440 IF Y$<>"Y" THEN 460
450 DISPLAY "->"&A$
460 IF YP$<>"Y" THEN 490
470 PRINT #3:">"&A$
480 REM DO NOT USE COMMAS IN INPUT
490 INPUT B$
500 PRINT #1:B$
510 IF YP$<>"Y" THEN 420
520 PRINT #3:B$
530 GOTO 420
540 REM TO CHANGE PITCH ON OTHER TERMINAL INPUT //PP SSS (PP=PITCH#, SSS=SLOPE#)
```

TALKING CALCULATOR

When the first personal computers appeared on the market, some skeptics called them "just expensive calculators." This program allows your computer to live up to that description, but it adds some features you won't find on most calculators. Electronic "music" and optional speech make it audibly more interesting than the normal calculator, with the added benefit that it can be used by the blind. You are given a choice of functions:

1 Addition
2 Multiplication
3 Subtraction
4 Division
5 Exponents
6 Square Root
7 Convert Kilograms to Pounds
8 Convert Centimeters to Inches
9 Retrieve Memory
10 Clear Memory

Also, by entering "11" at the function prompt, you can hear the computer speak the result of the most recent calculation. Entering the zero function ends the program. If you choose functions 1 through 4, you are asked to enter the first and second numbers you wish calculated. Entering zero as either the first or second number

enters the result of the last operation. Stationary sprites are used to simplify displaying the function menu while the calculations scroll up the screen like an adding machine tape.

When you see the "M" (Memory) prompt, press enter to add the result to the memory, enter "M" (Minus) to subtract if from memory. or enter "N" (No) to do neither. When you select a conversion, simply enter the metric quantity you want to convert to English measurements. If you need to display a result that is an extremely large number, you may choose function "99" for the Large Display Mode, then press R to display the last result or M to display the memory. (If you do not have the TI Speech Synthesizer attached, the computer will not perform the CALL SAY statements.)

```
100 REM *TALKING CALCULATOR* TI EXTENDED BASIC
110 REM BY DAVID MIGICOVSKY
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 CALL CLEAR :: CALL SCREEN(16)
150 FOR A=0 TO 12 :: CALL COLOR(A,13,1,14,16,16):: NEXT A
160 DISPLAY AT(10,9):" Calculator"
170 RANDOMIZE :: FOR A=1 TO 3 :: CALL SOUND(-100,110+2550*RND,0):: DISPLAY AT(16
,4):""
180 NEXT A :: DISPLAY AT(16,4):"Press Any Key to Begin" :: CALL KEY(0,K,S):: IF
S=0 THEN 170 :: CALL CLEAR
190 CALL CHAR(140,"07040404504848A8004040EA5F4A5F4A0040C04E484E42EE000040A02855F
4")
200 CALL CHAR(130,"000000E689C889860000004564544D44708808081122 42F90000404ADF4A5
FCA"):: CALL CHAR(135,"0046C949494949E607040407 50A8A8A8")
210 CALL CHAR(58,"0010007C001000004 0A0408FE80850300E0A0AE080C080E0"):: CALL CHAR
(62,"8080A6C9A9A7010600 4A9595955500000022 5544EE444444")
220 CALL SPRITE(#1,49,1,12,16,#2,50,1,23,16,#3,51,1,34,16,#4,52,1,45,16,#5,53,1,
56,16)
230 CALL SPRITE(#11,43,1,12,26,#12,120,1,23,26,#13,45,1,34,26,#14,58,1,45,26,#15
,60,1,56,27)
240 CALL SPRITE(#6,54,1,12,45,#7,55,1,23,45,#8,56,1,34,45,#9,57,1,45,45,#27,48,1
,71,29)
250 CALL SPRITE(#21,130,1,89,17,#22,142,1,105,17,#23,132,1,121,17)
260 CALL SPRITE(#16,59,1,12,56,#17,62,1,25,56,#18,63,1,36,56,#19,140,1,45,56,#28
,64,1,71,37)
270 C1=28 :: CALL SPRITE(#10,135,1,56,45,#20,136,1,56,56,#24,131,1,89,25,#25,141
,1,105,25,#26,133,1,121,25)
280 DISPLAY AT(1,1):"Functions" :: FOR A=1 TO C1 :: CALL COLOR(#A,13):: NEXT A :
: ON WARNING NEXT
290 CALL HCHAR(18,3,140,5):: ACCEPT AT(12,4)VALIDATE(DIGIT)SIZE(2)BEEP:C :: IF C
=99 THEN 610
300 IF C=11 THEN CALL SAY(STR$(REZ))! FUNCTION 11 SPEAKS RESULT
310 IF C>11 THEN GOSUB 590 :: GOTO 290 ELSE IF C<1 THEN 340
320 IF C>10 THEN 290
330 CALL COLOR(#27,1,#28,1):: FOR A=20 TO 1 STEP -1 :: CALL COLOR(#A,1):: NEXT A
 :: CALL COLOR(#C+10,13)
340 IF C<1 THEN GOSUB 580 ELSE 360
350 GOSUB 590 :: CALL KEY(0,K,S):: IF S=0 THEN 350 :: CALL SAY("GOODBYE"):: CALL
 CLEAR :: END
360 ON WARNING NEXT :: IF C>=9 THEN 440 ELSE IF C>5 THEN 370 :: ACCEPT AT(14,4)V
ALIDATE(NUMERIC)BEEP SIZE(10):F :: GOSUB 400
370 ON WARNING NEXT
380 FOR X=11 TO 14 :: CALL HCHAR(X,2,140,14):: NEXT X :: ACCEPT AT(16,4)VALIDATE
(NUMERIC)BEEP SIZE(10):S :: IF C=3 THEN S=-S
390 GOSUB 420 :: GOTO 440
```

```
400 FOR X=11 TO 16 :: CALL HCHAR(X,2,140,14):: NEXT X :: IF F=0 THEN 410 ELSE DI
SPLAY USING 490:F :: RETURN
410 IF F=0 THEN F=REZ :: DISPLAY USING 490:F :: RETURN
420 CALL HCHAR(16,2,140,14):: IF S=0 THEN 430 ELSE DISPLAY USING 490:S :: RETURN
430 IF S=0 THEN S=REZ :: DISPLAY USING 490:S :: RETURN
440 ON C GOSUB 500,510,500,520,530,540,550,560,570,600
450 ON WARNING NEXT :: C1=28 :: IF C)=9 THEN 280 ELSE CALL HCHAR(18,3,77):: ACCE
PT AT(18,3)VALIDATE("NM")SIZE(2)BEEP:M$
460 IF M$="N" THEN 280 ELSE IF M$="M" THEN 480 ELSE 470
470 MEM=MEM+REZ :: GOTO 280
480 MEM=MEM-REZ :: GOTO 280
490 IMAGE "             ############.###"
500 GOSUB 590 :: REZ=F+S :: DISPLAY USING 490:"---------------",REZ,"----------
----" :: RETURN
510 GOSUB 590 :: REZ=F*S :: DISPLAY USING 490:"---------------","           mult
",REZ,"---------------" :: RETURN
520 GOSUB 590 :: REZ=F/S :: DISPLAY USING 490:"---------------","            div
",REZ,"---------------" :: RETURN
530 GOSUB 590 :: REZ=F^S :: DISPLAY USING 490:"---------------","            exp
",REZ,"---------------" :: RETURN
540 GOSUB 590 :: REZ=SQR(S):: DISPLAY USING 490:"    square root",REZ,"---------
------" :: RETURN
550 GOSUB 590 :: REZ=S*2.201 :: DISPLAY USING 490:"          kg=",REZ,"---------
-lb---" :: RETURN
560 GOSUB 590 :: REZ=S/2.54 :: DISPLAY USING 490:"          cm=",REZ,"----------
in---" :: RETURN
570 GOSUB 590 :: CALL HCHAR(12,6,140):: DISPLAY USING 490:"      memory is",MEM,
"---------------" :: RETURN
580 CALL DELSPRITE(ALL):: CALL CLEAR :: DISPLAY AT(10,6):"Good-bye for now!" ::
DISPLAY AT(20,8):"Press any key"
590 CALL SOUND(-90,110+6050*RND,0):: CALL SOUND(150,110+3000*RND,5):: RETURN
600 GOSUB 590 :: MEM=0 :: CALL HCHAR(12,6,140,2):: DISPLAY USING 490:"memory is
clear","---------------" :: RETURN
610 CALL COLOR(#27,1,#28,1):: FOR X=20 TO 1 STEP -1 :: CALL COLOR(#X,1):: NEXT X
 :: DISPLAY AT(1,1):"" :: DISPLAY AT(2,6):"Large Display Mode"
620 ON WARNING NEXT :: ACCEPT AT(12,4)VALIDATE("MR")SIZE(2)BEEP:M$ :: PRINT
630 FOR X=8 TO 12 :: CALL HCHAR(X,4,140,3):: NEXT X
640 IF M$="M" THEN DISPLAY AT(23,15):USING "###############":MEM ELSE IF M$="R" T
HEN DISPLAY AT(23,15):USING "###############":REZ
650 REM CHANGE 600 IN NEXT LINE TO ADJUST DISPLAY TIME
660 FOR X=1 TO 600 :: NEXT X :: DISPLAY AT(23,1):"" :: DISPLAY AT(24,1):"" :: GO
TO 280
```

METRIC CONVERTER

The instructions are fairly self-explanatory in this convenient converter. Enter the number of the function you wish to perform at the "FCN" prompt. At the "->E" prompt, press Enter to convert to English or enter M to convert to Metric. Next, you simply enter the number to be converted. For example, to convert normal body temperature to Celsius, choose the first function, press Enter at the next prompt, then enter 98.6. The computer displays the result, 37 degrees Celsius. Selecting Function 9 prints the instructions page, while Function 0 stops the program. As in the Calculator program, sprites were used to display the functions quickly while the results scroll up the screen.

```
100 REM * METRIC CONVERTER * TI EXTENDED BASIC
110 REM BY DAVID MIGICOVSKY
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 CALL SCREEN(16):: CALL CLEAR :: FOR A=0 TO 12 :: CALL COLOR(A,5,1,14,16,16):
: NEXT A
150 DISPLAY AT(10,6):"Metric Converter" :: FOR A=1 TO 500 :: NEXT A
160 CALL DELSPRITE(ALL):: CALL CLEAR :: PRINT TAB(6);"METRIC CONVERTER": :
170 PRINT "TO CONVERT FROM METRIC TO   ENGLISH, PRESS ENTER AT THE ->E PROMPT. T
O CONVERT FROM ENGLISH TO METRIC, ENTER M."
180 PRINT :"    FUNCTIONS:": :" 1. CELSIUS/FAHRENHEIT":" 2. MILLILITERS/FLUID OU
NCES":" 3. GRAMS/OUNCES"
190 PRINT " 4. KILOGRAMS/POUNDS":" 5. CENTIMETERS/INCHES":" 6. METERS/FEET":" 7.
 KILOMETERS/MILES":" 8. LITERS/QUARTS"
200 PRINT " 9. THESE INSTRUCTIONS":" 0. END": :" PRESS ANY KEY TO CONTINUE"
210 CALL KEY(0,K,S):: IF S=0 THEN 210
220 CALL CLEAR
230 CALL CHAR(101,"000002025ZAAAAA900020505030105024040526553010502000000004A959
555")
240 CALL CHAR(130,"000000E689C8898600000045645440440")
250 CALL CHAR(105,"0000000050A8A8A8008080AAB5B5950008OCFEFFFE0C08")
260 CALL SPRITE(#1,49,1,12,16,#2,50,1,23,16,#3,51,1,34,16,#4,52,1,45,16)
270 CALL SPRITE(#11,67,1,12,26,#12,101,1,23,26,#13,102,1,37,23,#14,103,1,48,26)
280 CALL SPRITE(#5,53,1,12,45,#6,54,1,23,45,#7,55,1,34,45,#8,56,1,45,45)
290 CALL SPRITE(#21,130,1,89,17,#22,107,1,107,17,#23,35,1,121,17)
300 CALL SPRITE(#15,104,1,12,56,#16,105,1,23,56,#17,106,1,36,56,#18,76,1,45,56)
310 C1=28 :: CALL SPRITE(#10,32,1,56,45,#24,131,1,89,25,#25,69,1,106,27)
320 DISPLAY AT(1,1):"FUNCTIONS" :: FOR A=1 TO C1 :: CALL COLOR(#A,5):: NEXT A ::
 ON WARNING NEXT
330 ACCEPT AT(12,4)VALIDATE(DIGIT)SIZE(1)BEEP:C :: IF C=9 THEN 160 ELSE IF C=0 T
HEN 360
340 REM ERASE UNUSED FUNCTIONS
350 FOR A=18 TO 1 STEP -1 :: CALL COLOR(#A,1):: NEXT A :: CALL COLOR(#C+10,5)
360 IF C<>0 THEN 370 ELSE CALL DELSPRITE(ALL):: CALL CLEAR :: END
370 ON WARNING NEXT :: ACCEPT AT(14,4)BEEP VALIDATE("","M")SIZE(1):F$
380 FOR A=11 TO 16 :: CALL HCHAR(A,2,140,14):: NEXT A
390 ON WARNING NEXT :: FOR A=11 TO 14 :: CALL HCHAR(A,2,140,14):: NEXT A :: ACCE
PT AT(16,2)VALIDATE(NUMERIC)BEEP SIZE(10):S
400 CALL HCHAR(16,2,140,14):: DISPLAY USING 430:S
410 ON C GOSUB 440,470,500,520,540,560,580,610
420 C1=28 :: GOTO 320
430 IMAGE "            ##########.####"
440 IF F$<>"M" THEN 450 :: R=(S-32)*5/9 :: DISPLAY USING 430:R :: PRINT :: RETUR
N
450 R=(S*1.8)+32 :: DISPLAY USING 430:R :: PRINT :: RETURN
460 REM ** CANADIAN AND BRITISH USERS CHANGE 29.574 TO  28.41
470 IF F$<>"M" THEN 490 :: R=S*29.574 :: DISPLAY USING 430:R :: PRINT :: RETURN
480 REM ** CANADIAN AND BRITISH USERS CHANGE 0.0338134  TO 0.0351988
490 R=S*0.0338134 :: DISPLAY USING 430:R :: PRINT :: RETURN
500 IF F$<>"M" THEN 510 :: R=S*28.35 :: DISPLAY USING 430:R :: PRINT :: RETURN
510 R=S*0.0352733 :: DISPLAY USING 430:R :: PRINT :: RETURN
520 IF F$<>"M" THEN 530 :: R=S*0.4536 :: DISPLAY USING 430:R :: PRINT :: RETURN
530 R=S*2.2045855 :: DISPLAY USING 430:R :: PRINT :: RETURN
540 IF F$<>"M" THEN 550 :: R=S*2.54 :: DISPLAY USING 430:R :: PRINT :: RETURN
550 R=S*0.3937 :: DISPLAY USING 430:R :: PRINT :: RETURN
560 IF F$<>"M" THEN 570 :: R=S*0.3048 :: DISPLAY USING 430:R :: PRINT :: RETURN
570 R=S*3.280839 :: DISPLAY USING 430:R :: PRINT :: RETURN
580 IF F$<>"M" THEN 590 :: R=S*1.6093 :: DISPLAY USING 430:R :: PRINT :: RETURN
590 R=S*0.62138 :: DISPLAY USING 430:R :: PRINT :: RETURN
600 REM ** CANADIAN AND BRITISH USERS CHANGE 0.94635 TO  1.1365
610 IF F$<>"M" THEN 630 :: R=S*0.94635 :: DISPLAY USING 430:R :: PRINT :: RETURN
620 REM ** CANADIAN AND BRITISH USERS CHANGE 1.056691 TO 0.8798944
630 R=S*1.056691 :: DISPLAY USING 430:R :: PRINT :: RETURN
```

FRENCH NOUNS

Your computer can be your own private French tutor with this TI BASIC program. You are drilled and quizzed on three levels of French nouns. Actually, this program could be modified to help you learn just about anything requiring drill. Two sets of Data statements were used so that the proper French accent symbols could be displayed in the Review mode, while allowing you to use regular upper-case letters for your response in the Test section. Those with the 99/4 may enter all the data as upper-case and use the redefined symbols (@, #, $, ^) for the accented letters.

```
100 REM *FRENCH NOUNS* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM  COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL SCREEN(15)
140 CALL CLEAR
150 CALL CHAR(64,"0018207C4078407C")
160 CALL CHAR(35,"0030087C4078407C")
170 CALL CHAR(36,"0000003C40403C08")
180 CALL CHAR(94,"00300838447C4444")
190 CALL CHAR(123,"100810007C")
200 OPTION BASE 1
210 DIM W$(150,2)
220 GOSUB 1380
230 GOSUB 1020
240 CALL CLEAR
250 CALL COLOR(12,2,1)
260 CALL COLOR(13,2,1)
270 CALL COLOR(14,2,1)
280 PRINT : :"ENTER": :" 1. TO REVIEW WORDS": :" 2. TO TAKE TEST": :" 3. TO END
PROGRAM": :
290 S=0
300 INPUT "":C
310 IF C>3 THEN 300
320 IF C<1 THEN 300
330 IF C=3 THEN 950
340 PRINT : :"ENTER LEVEL: 1,2, OR 3": :
350 INPUT "":L
360 IF L=1 THEN 370 ELSE 390
370 X=0
380 GOTO 440
390 IF L=2 THEN 400 ELSE 420
400 X=50
410 GOTO 440
420 IF L=3 THEN 430 ELSE 440
430 X=100
440 IF L>3 THEN 340
450 IF L<1 THEN 340
460 CALL CLEAR
470 IF C=2 THEN 570
480 PRINT "WORD REVIEW, LEVEL ";L: : : :"PRESS ANY KEY FOR NEXT WORD": : : :"ENG
LISH","FRANCAIS"
490 PRINT "-------","--------": :
500 CALL HCHAR(22,21,123)
510 FOR I=X+1 TO X+25
520 PRINT :W$(I,1),W$(I,2)
530 CALL KEY(0,KEY,STATUS)
540 IF STATUS=0 THEN 530
550 NEXT I
560 GOTO 280
570 PRINT "TEST LEVEL ";L: : : :"SHALL I GIVE YOU THE WORDS-": :" 1. IN ENGLISH"
: :"    (YOU RESPOND IN FRENCH)"
```

```
580 PRINT :" 2. IN FRENCH": :"    (YOU RESPOND IN ENGLISH)": :
590 INPUT "":F
600 PRINT :"BE SURE YOUR SPELLING IS    EXACTLY CORRECT. USE UPPER- CASE LETTERS
 WITH NO ACCENTS": :
610 IF F=2 THEN 800
620 PRINT :"ENTER THE FRENCH WORD -":"YOU HAVE 2 TRIES": :
630 FOR I=X+26 TO X+50
640 PRINT :W$(I,1);"?": :
650 INPUT "":A$
660 IF A$=W$(I,2)THEN 730
670 PRINT :"WRONG--TRY AGAIN": :
680 INPUT "":A$
690 IF A$=W$(I,2)THEN 730
700 PRINT :"SORRY. THE ANSWER IS ";W$(I,2):
710 CALL SOUND(90,110,3)
720 GOTO 750
730 PRINT :"THAT'S RIGHT!"
740 S=S+4
750 NEXT I
760 PRINT : :"YOUR SCORE IS ";S;"%": : :
770 IF S>95 THEN 780 ELSE 280
780 GOSUB 1560
790 GOTO 280
800 PRINT :"ENTER THE ENGLISH WORD -":"YOU HAVE 2 TRIES": :
810 FOR I=X+26 TO X+50
820 PRINT :W$(I,2);"?": :
830 INPUT "":A$
840 IF A$=W$(I,1)THEN 910
850 PRINT :"WRONG--TRY AGAIN": :
860 INPUT "":A$
870 IF A$=W$(I,1)THEN 910
880 PRINT :"SORRY. THE ANSWER IS ";W$(I,1):
890 CALL SOUND(90,110,3)
900 GOTO 930
910 PRINT :"THAT'S RIGHT!"
920 S=S+4
930 NEXT I
940 GOTO 760
950 CALL CLEAR
960 PRINT TAB(10);"AU REVOIR": : :
970 GOSUB 1380
980 FOR D=1 TO 500
990 NEXT D
1000 CALL CLEAR
1010 END
1020 FOR I=1 TO 150
1030 FOR J=1 TO 2
1040 READ W$(I,J)
1050 NEXT J
1060 NEXT I
1070 REM LEVEL 1
1080 DATA Dog,Chien,Cat,Chat,Apple,Pomme,Fly,Mouche,Ice,Glace,Desk,Bureau,Street
,Rue
1090 DATA Beef,Boeuf,Bread,Pain,Ham,Jambon,Butter,Beurre,Fish,Poisson,Hat,Chapea
u,Fire,Feu,Water,Eau
1100 DATA House,Maison,Chair,Chaise,Boat,Bateau,Bird,Oiseau,Neighbor,Voisin
1110 DATA Pocket,Poche,War,Guerre,Peace,Paix,School,@cole,Church,@glise
1120 DATA DOG,CHIEN,CAT,CHAT,APPLE,POMME,FLY,MOUCHE,ICE,GLACE,DESK,BUREAU,STREET
,RUE
1130 DATA BEEF,BOEUF,BREAD,PAIN,HAM,JAMBON,BUTTER,BEURRE,FISH,POISSON,HAT,CHAPEA
U,FIRE,FEU,WATER,EAU
1140 DATA HOUSE,MAISON,CHAIR,CHAISE,BOAT,BATEAU,BIRD,OISEAU,NEIGHBOR,VOISIN
1150 DATA POCKET,POCHE,WAR,GUERRE,PEACE,PAIX,SCHOOL,ECOLE,CHURCH,EGLISE
1160 REM   LEVEL 2
1170 DATA Mother,M#re,Father,P#re,Sister,Soeur,Brother,Fr#re,Friend,Ami,Key,Clef
,Ticket,Billet,Man,Homme
```

```
1180 DATA Woman,Femme,Boy,Gar$on,Girl,Fille,Watch,Montre,Clock,Horloge,Oven,Four
,Potato,Pomme de terre
1190 DATA Rabbit,Lapin,Vegetable,L@gume,Light,Lumi#re,Iron,Fer,Steel,Acier,Book,
Livre,Record,Disque,Tape,Ruban
1200 DATA Pen,Stylo,Pencil,Crayon
1210 DATA MOTHER,MERE,FATHER,PERE,SISTER,SOEUR,BROTHER,FRERE,FRIEND,AMI,KEY,CLEF
,TICKET,BILLET
1220 DATA MAN,HOMME,WOMAN,FEMME,BOY,GARCON,GIRL,FILLE,WATCH,MONTRE,CLOCK,HORLOGE
,OVEN,FOUR
1230 DATA POTATO,POMME DE TERRE,RABBIT,LAPIN,VEGETABLE,LEGUME,LIGHT,LUMIERE,IRON
,FER,STEEL,ACIER
1240 DATA BOOK,LIVRE,RECORD,DISQUE,TAPE,RUBAN,PEN,STYLO,PENCIL,CRAYON
1250 REM LEVEL 3
1260 DATA Tape Recorder,Magn@tophone,Computer,Ordinateur,Record player,Tourne-di
sque,Flag,Drapeau
1270 DATA United States of America,@tats-Unis d'Am@rique,Knowledge,Science,Vacuu
m Cleaner,Aspirateur
1280 DATA Sink,Lavabo,Bath,Bain,Shower,Douche,Clock Radio,Radio-r@veil,Paper,Pap
ier,Paper towel,Essuie-tout
1290 DATA Soap,Savon,Detergent,D@tersif,Razor,Rasoir,Shaving cream,Cr#me ^ barbe
,Screwdriver,Tournevis
1300 DATA Hammer,Marteau,Nail,Clou,Tool,Outil,Bed,Lit,Pillow,Oreiller,Flower,Fle
ur,End,Fin
1310 DATA TAPE RECORDER,MAGNETOPHONE,COMPUTER,ORDINATEUR,RECORD PLAYER,TOURNE-DI
SQUE,FLAG,DRAPEAU
1320 DATA UNITED STATES OF AMERICA,ETATS-UNIS D'AMERIQUE,KNOWLEDGE,SCIENCE,VACUU
M CLEANER,ASPIRATEUR
1330 DATA SINK,LAVABO,BATH,BAIN,SHOWER,DOUCHE,CLOCK RADIO,RADIO-REVEIL,PAPER,PAP
IER,PAPER TOWEL,ESSUIE-TOUT
1340 DATA SOAP,SAVON,DETERGENT,DETERSIF,RAZOR,RASOIR,SHAVING CREAM,CREME A BARBE
,SCREWDRIVER,TOURNEVIS
1350 DATA HAMMER,MARTEAU,NAIL,CLOU,TOOL,OUTIL,BED,LIT,PILLOW,OREILLER,FLOWER,FLE
UR,END,FIN
1360 RETURN
1370 REM FLAG
1380 T1$="FRENCH"
1390 T2$="NOUNS "
1400 CALL COLOR(12,1,1)
1410 CALL COLOR(13,1,1)
1420 CALL COLOR(14,1,1)
1430 FOR I=1 TO 3
1440 CALL VCHAR(7,11+I,127,7)
1450 CALL VCHAR(7,14+I,128,7)
1460 CALL VCHAR(7,17+I,140,7)
1470 NEXT I
1480 CALL COLOR(12,5,5)
1490 CALL COLOR(13,16,16)
1500 CALL COLOR(14,7,7)
1510 FOR I=1 TO 6
1520 CALL HCHAR(5,13+I,ASC(SEG$(T1$,I,1)))
1530 CALL HCHAR(16,13+I,ASC(SEG$(T2$,I,1)))
1540 NEXT I
1550 REM LA MARSEILLAISE
1560 CALL SOUND(110,196,0)
1570 CALL SOUND(300,262,0)
1580 CALL SOUND(300,262,0)
1590 CALL SOUND(300,294,0)
1600 CALL SOUND(300,294,0)
1610 CALL SOUND(500,392,0)
1620 CALL SOUND(150,330,0)
1630 CALL SOUND(400,262,0)
1640 RETURN
```

FRENCH TEACHER

If you have the TI Speech Synthesizer and the Terminal Emulator II command module, you may want to take a break from studying TI BASIC to learn a little French. Some words are difficult to pronounce with the text-to-speech feature of TE2 if you type the phrase in English. This program utilizes the allophone feature of TE2, which allows you to break words into individual parts and specify the allophone numbers that you wish to have spoken. In this program, the allophone numbers are read from strings stored in Data statements. The first character of the number string indicates the number of syllables in the word. Each group of three digits following that make up an allophone number. Remember that the allophones in TE2 were recorded in English using standard rules of American English pronunciation. Thus, your computer may not be able to vocalize like a true Parisian, but this program should be an entertaining introduction to French.

When you run the program, you will be asked if you would like to study French numbers or if you would prefer to learn names of months and the days of the week. When you review either section, you may press S (for "Speak") to hear a word spoken. After your review, you may take a quiz to test your progress. You may wish to adapt this program to teach other types of foreign language words or English as a second language. The use of allophones should provide you with a wide range of possibilities in using synthesized speech with computer assisted instruction programs.

```
100 REM   *FRENCH TEACHER* TI BASIC WITH SPEECH
110 REM  FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM  COPYRIGHT (C) 1983 BY STEVE DAVIS
130 REM   TERMINAL EMULATOR 2 AND SPEECH SYNTHESIZER REQUIRED
140 RANDOMIZE
150 DIM X$(28),Z$(28),W(20)
160 CALL SCREEN(15)
170 GOSUB 1370
180 OPEN #1:"ALPHON",INTERNAL
190 OPEN #2:"SPEECH",OUTPUT
200 PRINT #2://35 160"
210 CALL CLEAR
220 C$=CHR$(86)&CHR$(15)&CHR$(80)&CHR$(102)&CHR$(46)&CHR$(82)
230 PRINT "BONJOUR!":
240 PRINT #1:C$
250 PRINT :::"WHICH DO YOU WANT TO STUDY?"::"1 - NUMBERS"::"2 - MONTHS AND DAYS"
::"3 - END PROGRAM":::
260 FOR X=1 TO 20
270 W(X)=0
280 NEXT X
290 SC=0
300 INPUT "":C1
310 IF C1=3 THEN 2400
320 IF C1>2 THEN 250
330 IF C1<1 THEN 250
340 CALL CLEAR
350 PRINT :"ENTER"::"1 - TO REVIEW LESSON"::"2 - TO TAKE QUIZ":::
360 INPUT "":C2
370 CALL CLEAR
380 IF C1=2 THEN 420
```

```
390 X=28
400 RESTORE 740
410 IF C2=2 THEN 970 ELSE 450
420 X=19
430 RESTORE 860
440 IF C2=2 THEN 2000 ELSE 1740
450 PRINT "YOU WILL LEARN TO SPELL";"NUMBERS IN FRENCH.";:;"TO PRACTICE PRONUNCIA
TON,"
460 PRINT "PRESS S AFTER EACH WORD AND";"THE COMPUTER WILL SPEAK IT.";:;"PRESS EN
TER FOR THE NEXT    WORD"
470 PRINT ;:;"#";TAB(10);"FRANCAIS";"-";TAB(10);"--------"
480 CALL HCHAR(23,16,151)
490 PRINT ::
500 RESTORE 740
510 FOR I=1 TO X
520 READ N,F$,AL$
530 N$=STR$(N)
540 PRINT :N$;TAB(10);F$
550 CALL KEY(0,KEY,STATUS)
560 IF STATUS=0 THEN 550
570 IF KEY=13 THEN 710
580 IF KEY<>83 THEN 550
590 REM   READ ALLOPHONES
600 A=VAL(SEG$(AL$,1,1))
610 B$=CHR$(250)&CHR$(255)&CHR$(A)
620 FOR J=2 TO LEN(AL$)STEP 3
630 C=VAL(SEG$(AL$,J,3))
640 B$=B$&CHR$(C)
650 NEXT J
660 PRINT #1:B$
670 CALL KEY(0,KEY,STATUS)
680 IF STATUS=0 THEN 670
690 IF KEY=83 THEN 660
700 IF KEY<>13 THEN 670
710 NEXT I
720 GOTO 250
730 REM  DATA FOR NUMBERS
740 DATA 1,UN,1040081,2,DEUX,1089040,3,TROIS,1112082083001
750 DATA 4,QUATRE,1106026113082018,5,CINQ,1120026080105,6,SIX,1120032121
760 DATA 7,SEPT,1120034113,8,HUIT,1084032113,9,NEUF,1078040116
770 DATA 10,DIX,1088032121,11,ONZE,1067080100,12,DOUZE,1088068100
780 DATA 13,TREIZE,1112082056100,14,QUATORZE,2107026112066100,15,QUINZE,11070470
79100
790 DATA 16,SEIZE,2120034100,17,DIX-SEPT,2088032120034113,18,DIX-HUIT,2088032099
083032113
800 DATA 19,DIX-NEUF,20880320990780401l6,20,VINGT,1097047080
810 DATA 30,TRENTE,1112082027079112,40,QUARANTE,21040260820270791l2,50,CINQUANTE
,21200260791030270791l2
820 DATA 60,SOIXANTE,212008301812002707911 2,70,SOIXANTE-DIX,3120083018120027079l
13089032121
830 DATA 80,QUATRE-VINGTS,3104026113082069097026080
840 DATA 90,QUATRE-VINGT-DIX,41040261130820690970260800880321 21,100,CENT,1120052
079
850 REM  DATA FOR MONTHS/DAYS
860 DATA MONDAY,LUNDI,207304007908803 2,TUESDAY,MARDI,2076029089032
870 DATA WEDNESDAY,MERCREDI,3076057107082012088053,THURSDAY,JEUDI,2102040088053
880 DATA FRIDAY,VENDREDI,309704807908808201208805 3,SATURDAY,SAMEDI,3120026077012
088053
890 DATA SUNDAY,DIMANCHE,208803207605208012 3
900 DATA JANUARY,JANVIER,3102052079098032036,FEBRUARY,FEVRIER,3115034098082032036
910 DATA MARCH,MARS,1076029100,APRIL,AVRIL,2026098082060074
920 DATA MAY,MAI,1076036,JUNE,JUIN,1102083026080,JULY,JUILLET,2102083032085034
930 DATA AUGUST,A0@T,1046,SEPTEMBER,SEPTEMBRE,212003411011304807708708201 9
940 DATA OCTOBER,OCTOBRE,204810511204808708201 9,NOVEMBER,NOVEMBRE,20780430970480
77087082019
```

```
950 DATA DECEMBER,DECEMBRE,20880361200480770870820l9
960 REM   NUMBER QUIZ
970 PRINT :"CHOOSE":::"1. ANSWER WITH NUMBERS":"   (PROMPT IS IN FRENCH)"::"2. A
NSWER IN FRENCH"
980 PRINT "   (PROMPT IS NUMERIC)"::
990 INPUT "":C3
1000 IF C3<1 THEN 990
1010 IF C3>2 THEN 990
1020 PRINT
1030 FOR I=1 TO X
1040 READ X$(I),Z$(I),ZX$
1050 NEXT I
1060 FOR I=1 TO 20
1070 X=(INT(RND*28)+1)
1080 FOR II=1 TO 20
1090 IF W(II)=X THEN 1070
1100 NEXT II
1110 W(I)=X
1120 IF C3=1 THEN 1160
1130 C$=X$(W(I))
1140 D$=Z$(W(I))
1150 GOTO 1180
1160 C$=Z$(W(I))
1170 D$=X$(W(I))
1180 C$=STR$(I)&". "&C$&"? "
1190 INPUT C$:C1$
1200 IF C1$=D$ THEN 1270
1210 PRINT :"WRONG, TRY AGAIN"::
1220 INPUT "":C1$
1230 IF C1$=D$ THEN 1270
1240 PRINT :"SORRY. ANSWER IS ";D$::
1250 CALL SOUND(100,110,0)
1260 GOTO 1290
1270 SC=SC+1
1280 PRINT :"THAT'S RIGHT!"::
1290 NEXT I
1300 SC=SC*5
1310 PRINT "SCORE IS";SC;"%"::
1320 IF SC>90 THEN 1340
1330 GOTO 250
1340 GOSUB 1610
1350 GOTO 250
1360 REM  FLAG
1370 CALL CLEAR
1380 T1$=" FRENCH "
1390 T2$="FRANCAIS"
1400 CALL CHAR(150,"10081")
1410 CALL CHAR(151,"100810007C")
1420 CALL CHAR(64,"1028004444444438")
1430 CALL COLOR(9,1,1)
1440 CALL COLOR(10,1,1)
1450 CALL COLOR(11,1,1)
1460 FOR I=1 TO 3
1470 CALL VCHAR(8,I+11,96,7)
1480 CALL VCHAR(8,I+14,104,7)
1490 CALL VCHAR(8,17+I,112,7)
1500 NEXT I
1510 CALL COLOR(9,5,5)
1520 CALL COLOR(10,16,16)
1530 CALL COLOR(11,7,7)
1540 FOR I=1 TO 8
1550 CALL HCHAR(5,12+I,ASC(SEG$(T1$,I,1)))
1560 CALL HCHAR(17,12+I,ASC(SEG$(T2$,I,1)))
1570 IF I=5 THEN 1580 ELSE 1590
1580 CALL HCHAR(18,I+12,150)
```

```
1590 NEXT I
1600 REM  LA MARSEILLAISE
1610 CALL SOUND(110,196,10)
1620 CALL SOUND(300,262,10)
1630 CALL SOUND(300,262,10)
1640 CALL SOUND(300,294,10)
1650 CALL SOUND(300,294,10)
1660 CALL SOUND(500,392,10)
1670 CALL SOUND(150,330,10)
1680 CALL SOUND(400,262,10)
1690 IF SC=100 THEN 1720
1700 FOR DELAY=1 TO 300
1710 NEXT DELAY
1720 RETURN
1730 REM   STUDY DAYS/MONTHS
1740 PRINT "YOU WILL LEARN TO SPELL":"DAYS AND MONTHS IN FRENCH.":::"TO PRACTICE
PRONUNCIATON,"
1750 PRINT "PRESS S AFTER EACH WORD AND":"THE COMPUTER WILL SPEAK IT.":::"PRESS E
NTER FOR THE NEXT    WORD"
1760 PRINT :::"ENGLISH";TAB(11);"FRANCAIS":"--------";TAB(11);"---------"
1770 CALL HCHAR(23,17,151)
1780 PRINT ::
1790 FOR I=1 TO X
1800 READ F$,J$,AL$
1810 PRINT :F$;TAB(11);J$
1820 CALL KEY(0,KEY,STATUS)
1830 IF STATUS=0 THEN 1820
1840 IF KEY=13 THEN 1970
1850 IF KEY<>83 THEN 1820
1860 A=VAL(SEG$(AL$,1,1))
1870 B$=CHR$(250)&CHR$(255)&CHR$(A)
1880 FOR J=2 TO LEN(AL$)STEP 3
1890 C=VAL(SEG$(AL$,J,3))
1900 B$=B$&CHR$(C)
1910 NEXT J
1920 PRINT #1:B$
1930 CALL KEY(0,KEY,STATUS)
1940 IF STATUS=0 THEN 1930
1950 IF KEY=83 THEN 1920
1960 IF KEY<>13 THEN 1930
1970 NEXT I
1980 GOTO 250
1990 REM  DAY/MONTH QUIZ
2000 PRINT :"CHOOSE":::"1. ANSWER IN ENGLISH"::"2. ANSWER IN FRENCH"::
2010 INPUT "":C3
2020 IF C3<1 THEN 2010
2030 IF C3>2 THEN 2010
2040 PRINT
2050 FOR I=1 TO X
2060 READ X$(I),Z$(I),ZX$
2070 NEXT I
2080 FOR I=1 TO 15
2090 X=(INT(RND*19)+1)
2100 FOR II=1 TO 15
2110 IF W(II)=X THEN 2090
2120 NEXT II
2130 W(I)=X
2140 IF C3=1 THEN 2200
2150 C$=X$(W(I))
2160 D$=Z$(W(I))
2170 IF D$<>"AO@T" THEN 2190
2180 D$="AOUT"
2190 GOTO 2220
2200 C$=Z$(W(I))
2210 D$=X$(W(I))
```

```
2220 C$=STR$(I)&". "&C$&"? "
2230 INPUT C$:C1$
2240 IF C1$=D$ THEN 2310
2250 PRINT :"WRONG, TRY AGAIN"::
2260 INPUT "":C1$
2270 IF C1$=D$ THEN 2310
2280 PRINT :"SORRY. ANSWER IS ";D$::
2290 CALL SOUND(100,110,0)
2300 GOTO 2330
2310 SC=SC+1
2320 PRINT :"THAT'S RIGHT!"::
2330 NEXT I
2340 SC=INT(SC/15*100)
2350 PRINT "SCORE IS";SC;"%"::
2360 IF SC>90 THEN 2380
2370 GOTO 250
2380 GOSUB 1610
2390 GOTO 250
2400 CALL CLEAR
2410 C$=CHR$(15)&CHR$(82)&CHR$(13)&CHR$(98)&CHR$(83)&CHR$(29)
2420 PRINT TAB(10);"AU  REVOIR"::::
2430 PRINT #1:C$
2440 GOSUB 1380
2450 CALL CLEAR
2460 END
```

MORSE CODER

Long before the ASCII Code was created, another code, the International Morse Code, was used for data transmission, and it is still in use today for many applications. This program is an example of what is referred to as C.A.I., or Computer Assisted Instruction. Not only does it teach you the code, it is also a translator. TI Extended BASIC was used so that the program could be structured into a series of subprograms that can be called by selecting menu options. Within these subprograms, "flags" are set to tell the computer if you are using a printer or wish to hear the code sounded. (Select the Print Code option ONLY if you have a printer connected.)

When you want the computer to Decode, you may use either the Underline character ("_", Function U on the 99/4A) or a Comma for the "Dash." A Period (".") is used for the "Dot." In the Test Mode, you must use the Underline for the dash. In Decode, the Number symbol ("#") will be returned if you make a typing error, indicating a translation cannot be made.

One set of Data statements is used for all modes of the program, so you could conceivably change the Read and Data statements to use the program to teach another kind of code or language. The Test mode randomly selects 10 different characters for you to either code or decode, then it gives you your score. The Print Code Table option prints the entire code on your printer to give you a reference chart. The Sounder routine plays the dots at a higher pitch than the dashes to make them easy to distinguish.

```
100 REM *MORSE CODER* TI EXTENDED BASIC
110 REM BY MIKE WILCOX
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 RANDOMIZE :: CALL CHAR(139,"F0F0F0F00F0F0F0F"):: CALL CLEAR :: CALL SCREEN(1
6)
150 FOR I=1 TO 32 :: CALL HCHAR(1,I,139):: NEXT I
160 FOR I=1 TO 24 :: CALL VCHAR(I,32,139):: NEXT I
170 FOR I=32 TO 1 STEP -1 :: CALL HCHAR(24,I,139):: NEXT I
180 FOR I=24 TO 1 STEP -1 :: CALL VCHAR(I,1,139):: NEXT I
190 DISPLAY AT(3,7):"PROGRAMS FOR THE" :: DISPLAY AT(5,7):"TI HOME COMPUTER"
200 DISPLAY AT(7,11):"PRESENTS:" :: DISPLAY AT(10,4):"COMPUTER COURSE IN THE" ::
 DISPLAY AT(12,3):"INTERNATIONAL MORSE CODE"
210 DISPLAY AT(23,8):"COPYRIGHT 1982" :: DISPLAY AT(18,4):"PRESS ANY KEY TO BEGI
N"
220 DEF XX=INT(14*RND+2)
230 CALL KEY(0,K,S):: CALL SOUND(22,2975,0):: CALL COLOR(14,XX,1)
240 T=850*INT(RND*2):: CALL SOUND(22,2125+T,0):: CALL COLOR(14,1,XX):: CALL SOUN
D(-99,2125,0):: IF S=0 THEN 230
250 OPTION BASE 1 :: DIM U(52):: CALL SCREEN(8)
260 CALL CLEAR
270 DISPLAY AT(2,8)BEEP:" MENU:" :: DISPLAY AT(4,1):"PRESS-" :: DISPLAY AT(7,1):
"1. TO DISPLAY CODE TABLES"
280 DISPLAY AT(9,1):"2. TO PRINT CODE TABLES" :: DISPLAY AT(11,1):"3. TO CODE ME
SSAGE" :: DISPLAY AT(13,1):"4. TO DECODE MESSAGES"
290 DISPLAY AT(15,1):"5. TO TEST YOURSELF ON CODE" :: DISPLAY AT(17,1):"6. END P
ROGRAM"
300 CALL KEY(0,K,S):: W=RND :: IF S=0 THEN 300
310 IF (K(49)+(K)54)THEN 300
320 CALL CLEAR
330 ON K-48 GOTO 340,350,370,380,390,400
340 PR$="" :: FLAG=0 :: CALL CODE(FLAG,PR$):: GOTO 260
350 DISPLAY "ENTER PRINTER NAME DEVICE" :: INPUT "NAME:":PR$
360 FLAG=1 :: CALL CODE(FLAG,PR$):: GOTO 260
370 CALL CODER :: GOTO 260
380 CALL DECODER :: GOTO 260
390 CALL TEST(U()):: GOTO 260
400 CALL CLEAR :: STOP
410 SUB CODE(FLAG,PR$)
420 CALL CLEAR
430 IF FLAG()1 THEN 450
440 OPEN #1:PR$,OUTPUT
450 C=1 :: R=1
460 RESTORE 900
470 DISPLAY AT(1,2):"INTERNATIONAL MORSE CODE:"
480 FOR I=1 TO 26
490 READ L$,CODE$
500 R=R+2
510 IF R)15 THEN R=3 :: C=C+7
520 DISPLAY AT(R,C):L$;" ";CODE$
530 NEXT I
540 CALL CHAR(97,RPT$("0",9)&"408102"):: CALL CHAR(98,RPT$("0",10)&"205408"):: C
ALL CHAR(99,RPT$("0",12)&"6666")
550 R=20 :: C=4
560 DISPLAY AT(18,3):"FOREIGN LANGUAGE LETTERS:"
570 FOR I=1 TO 6
580 READ L$,CODE$
590 DISPLAY AT(R,C):L$;" ";CODE$
600 R=R+3
610 IF R)23 THEN R=20 :: C=C+8
620 NEXT I
630 DISPLAY AT(19,4):"a" :: DISPLAY AT(22,4):"c" :: DISPLAY AT(19,12):"a" :: DIS
PLAY AT(22,12):"b" :: DISPLAY AT(19,20):"c" :: DISPLAY AT(22,20):"c"
640 IF FLAG()1 THEN 680
650 FOR I=1 TO 24 :: FOR J=1 TO 32 :: CALL GCHAR(I,J,X):: CALL HCHAR(I,J,30):: C
ALL HCHAR(I,J,X)
660 P$=P$&CHR$(X):: NEXT J :: PRINT #1:P$ :: P$="" :: NEXT I
```

```
670 GOTO 690
680 CALL KEY(0,K,S):: IF S=0 THEN 680
690 CALL CLEAR
700 DISPLAY AT(1,2):"INTERNATIONAL MORSE CODE:"
710 DISPLAY AT(3,1):"PUNCTUATION:"
720 R=5 :: C=1
730 FOR I=1 TO 10
740 READ L$,CODE$
750 DISPLAY AT(R,C):L$ :: DISPLAY AT(R,C+14):CODE$
760 R=R+2
770 NEXT I
780 DISPLAY AT(3,21):"NUMBERS:"
790 R=5 :: C=22
800 FOR I=1 TO 10
810 READ L$,CODE$
820 DISPLAY AT(R,C):L$;" ";CODE$
830 R=R+2
840 NEXT I
850 IF FLAG()1 THEN 890
860 FOR I=1 TO 24 STEP 2 :: FOR J=1 TO 32 :: CALL GCHAR(I,J,X):: CALL HCHAR(I,J,
30):: CALL HCHAR(I,J,X)
870 P$=P$&CHR$(X):: NEXT J :: PRINT #1:P$ :: P$="" :: NEXT I :: CLOSE #1
880 FLAG=0 :: SUBEXIT
890 CALL KEY(0,K,S):: IF S=0 THEN 890
900 DATA A,._,B,_..., C,_._., D,_.., E,., F,.._., G,__., H,...., I,.., J,.___, K,_._, L,._
..,M,__,N,_.,O,___
910 DATA P,.__., Q,__._, R,._., S,..., T,_,U,.._,V,..._,W,.__,X,_.._,Y,_.__,Z,__..
920 DATA A,.__._,A,._._,E,.._..,N,__.__,O,___.,U,..__
930 DATA ", COMMA",__..__,. PERIOD,._._._,? QUESTION,..__..,; SEMI-COLON,_._._.,
: COLON,___...
940 DATA ' APOSTROPHE,.____.,- HYPHEN,_...._,/ SLASH,_.._.,()PARENTHESIS,_.__._,
_ UNDERLINE,..__._
950 DATA 1,.____,2,..___,3,...__,4,...._,5,.....,6,_....,7,__...,8,___..,9,____.
,0,_____,
960 SUBEND
970 SUB CODER
980 DISPLAY AT(10,1)BEEP ERASE ALL:"WOULD YOU LIKE YOUR CODED": :"MESSAGES PRINT
ED-OUT (Y/N)?"
990 CALL KEY(3,K,S):: IF (K()89)*(K()78)THEN 990
1000 IF K=78 THEN FLAG=0 ELSE FLAG=1
1010 IF FLAG=0 THEN 1040
1020 DISPLAY "ENTER PRINTER NAME DEVICE" :: INPUT "NAME:":PR$
1030 OPEN #3:PR$,OUTPUT
1040 DISPLAY AT(10,1)BEEP ERASE ALL:"WOULD YOU LIKE YOUR CODED": :"MESSAGE SOUND
ED-OUT (Y/N)?"
1050 CALL KEY(3,K,S):: IF K()78 AND K()89 THEN 1050
1060 IF K=89 THEN FLAG2=1 ELSE FLAG2=0
1070 ROW=2 :: COL=1
1080 DISPLAY AT(2,1)ERASE ALL:"YOUR ARE IN THE CODING MODE "
1090 DISPLAY AT(10,3):"TO RETURN TO THE MENU ": :"  TYPE AND ENTER ""MENU""""
1100 DISPLAY AT(16,3):"PRESS ANY KEY TO BEGIN." :: CALL KEY(0,K,S):: IF S=0 THEN
 1100
1110 DISPLAY AT(10,10)ERASE ALL:"CODER:" :: DISPLAY AT(12,1)BEEP:"ENTER YOUR MES
SAGE TO CODE:": :"(TYPE AND ENTER ""MENU"" TO": :"RETURN.)"
1120 LINPUT MSG$
1130 IF MSG$="MENU" THEN 1430
1140 M$(1)=SEG$(MSG$,1,60):: M$(2)=SEG$(MSG$,61,120):: M$(3)=SEG$(MSG$,121,180):
: M$(4)=SEG$(MSG$,181,240):: M$(5)=SEG$(MSG$,241,300)
1150 CALL CLEAR :: DISPLAY AT(1,3):"TRANSLATING IN PROGRESS"
1160 FOR I=1 TO 5
1170 FOR J=1 TO LEN(M$(I))
1180 RESTORE 900
1190 A$=SEG$(M$(I),J,1):: IF A$=" " THEN CODE$=" " :: GOTO 1250
1200 FOR X=1 TO 52
1210 READ L$,CODE$
```

```
1220 IF A$=SEG$(L$,1,1)THEN 1250
1230 NEXT X
1240 IF A$<>L$ THEN CODE$="#"
1250 CMSG$=CMSG$&CODE$&" "
1260 FOR Z=1 TO LEN(CMSG$)
1270 COL=COL+1 :: IF COL<=31 THEN 1280 ELSE ROW=ROW+2 :: COL=2
1280 IF ROW>22 THEN ROW=3
1290 CALL HCHAR(ROW,COL,ASC(SEG$(CMSG$,Z,1)))
1300 NEXT Z
1310 CMSG$=""
1320 NEXT J
1330 NEXT I
1340 DISPLAY AT(1,1):" " :: DISPLAY AT(24,4):"TRANSLATION COMPLETED"
1350 IF FLAG=0 THEN 1410
1360 P$="" :: FOR X=2 TO 24 STEP 2 :: FOR Z=1 TO 32 :: CALL GCHAR(X,Z,M)
1370 P$=P$&CHR$(M)
1380 NEXT Z :: IF P$=RPT$(" ",32)THEN 1400
1390 PRINT #3:P$ :: P$="" :: NEXT X
1400 REM PRINTING COMPLETED
1410 IF FLAG2=1 THEN CALL SOUNDER(2,2)
1420 CALL KEY(0,K,S):: IF S=0 THEN 1420 ELSE ROW=2 :: COL=1 :: GOTO 1110
1430 IF FLAG=0 THEN 1450
1440 CLOSE #3
1450 FLAG=0 :: PR$="" :: SUBEND
1460 SUB DECODER :: MSG$=""
1470 DISPLAY AT(10,1)BEEP ERASE ALL:"WOULD YOU LIKE THE DECODED": :"MESSAGES PRI
NTED-OUT (Y/N)?"
1480 CALL KEY(3,K,S):: IF (K<>78)*(K<>89)THEN 1480
1490 IF K=78 THEN FLAG=0 ELSE FLAG=1
1500 IF FLAG=0 THEN 1530
1510 DISPLAY "ENTER PRINTER NAME DEVICE" :: INPUT "NAME:":PR$
1520 OPEN #4:PR$,OUTPUT
1530 DISPLAY AT(10,1)BEEP ERASE ALL:"WOULD YOU LIKE THE CODED": :"MESSAGE SOUNDE
D-OUT (Y/N)?"
1540 CALL KEY(3,K,S):: IF K<>78 AND K<>89 THEN 1540
1550 IF K=89 THEN FLAG2=1 ELSE FLAG2=0
1560 DISPLAY AT(2,1)ERASE ALL:"YOU ARE IN THE DECODING MODE" :: DISPLAY AT(5,1):
"PRESS ENTER FOR TRANSLATION"
1570 DISPLAY AT(7,6):"""M"" TO RETURN TO MENU": :"FOR YOUR CONVENIENCE YOU": :"M
AY USE A COMMA "","" TO PRINT"
1580 DISPLAY AT(13,1):"A DASH ""_"" (OR USE FCTN U)"
1590 DISPLAY AT(15,1):"YOU MUST LEAVE ONE SPACE": :"BETWEEN EACH LETTER, AND TWO
": :"SPACES BETWEEN EACH WORD."
1600 DISPLAY AT(22,3):"PRESS ANY KEY TO BEGIN"
1610 CALL KEY(0,K,S):: IF S=0 THEN 1610
1620 DISPLAY AT(4,10)BEEP ERASE ALL:"DECODER:" :: DISPLAY AT(6,1):"PRESS ENTER T
O TRANSLATE": :"PRESS M TO RETURN": :"PRESS C TO CORRECT"
1630 DISPLAY AT(12,1):"ENTER YOUR MESSAGE:"
1640 CALL KEY(3,K,S):: IF (K<>13)*(K<>32)*(K<>44)*(K<>46)*(K<>67)*(K<>77)*(K<>95
)THEN 1640
1650 IF K=67 AND MSG$="" THEN 1710 ELSE IF K=67 THEN MSG$=SEG$(MSG$,1,LEN(MSG$)-
1):: GOTO 1710
1660 IF K=77 THEN 2010
1670 IF K=44 THEN K=95
1680 IF K=32 THEN CALL SOUND(111,1000,5,2000,4,3000,3)
1690 IF K=13 THEN 1730
1700 MSG$=MSG$&CHR$(K)
1710 DISPLAY AT(14,1):MSG$
1720 GOTO 1640
1730 IF FLAG2=1 THEN CALL SOUNDER(14,1)
1740 CALL CLEAR :: ROW=2 :: COL=2 :: T$="" :: CMSG$=""
1750 DISPLAY AT(1,3):"TRANSLATING IN PROGRESS"
1760 S=POS(MSG$," ",1):: IF S=1 THEN T$=" " :: GOTO 1860
1770 IF S=0 THEN A$=MSG$ :: GOTO 1790
1780 A$=SEG$(MSG$,1,S-1)
```

```
1790 RESTORE 900
1800 FOR I=1 TO 52
1810 READ L$,CODE$
1820 IF A$=CODE$ THEN 1850
1830 NEXT I
1840 T$="#" :: GOTO 1860
1850 T$=SEG$(L$,1,1)
1860 FOR Z=1 TO LEN(T$)
1870 COL=COL+1
1880 IF COL<=31 THEN 1890 ELSE ROW=ROW+2 :: COL=2
1890 CALL HCHAR(ROW,COL,ASC(SEG$(T$,Z,1)))
1900 IF S=0 THEN 1930
1910 NEXT Z
1920 MSG$=SEG$(MSG$,S+1,LEN(MSG$)):: GOTO 1760
1930 DISPLAY AT(1,1):" " :: DISPLAY AT(24,3)BEEP:"TRANSLATION COMPLETED"
1940 IF FLAG=0 THEN 2000
1950 P$="" :: FOR X=2 TO 24 STEP 2 :: FOR Z=1 TO 32 :: CALL GCHAR(X,Z,M)
1960 P$=P$&CHR$(M)
1970 NEXT Z :: IF P$=RPT$(" ",32)THEN 1990
1980 PRINT #4:P$ :: P$="" :: NEXT X
1990 REM PRINTING COMPLETED
2000 CALL KEY(0,K,S):: IF S=0 THEN 2000 ELSE MSG$="" :: GOTO 1620
2010 IF FLAG=0 THEN 2030
2020 CLOSE #4
2030 SUBEND
2040 SUB TEST(U())
2050 SCORE,RIGHT=0 :: FOR I=1 TO 52 :: U(I)=0 :: NEXT I
2060 DISPLAY AT(1,5)ERASE ALL:"SELF-TEST MODE:"
2070 DISPLAY AT(3,1):"THE COMPUTER WILL RANDOMLY": :"SELECT 10 LETTERS, NUMBERS,
": :"OR PUNCTUATION SYMBOLS [NO"
2080 DISPLAY AT(9,1):"FOREIGN LANGUAGE LETTERS]": :"AND YOU WILL NEED TO PROVIDE
": :"EITHER THE CORRECT CODE, OR"
2090 DISPLAY AT(15,1):"THE LETTER, NUMBER OR SYMBOL": :"THAT IS BEING DEFINED."
2100 DISPLAY AT(19,1):"YOU WILL RECIEVE A SCORE AT": :"THE END OF THIS TEST.": :
"  PRESS ANY KEY TO START"
2110 CALL KEY(0,K,S):: IF S=0 THEN 2110
2120 DISPLAY AT(10,9)ERASE ALL:"TEST SET-UP" :: DISPLAY AT(12,9):"IN PROGRESS...
" :: DISPLAY AT(14,7):"PLEASE STAND BY..."
2130 REM SET-UP TEST
2140 FOR I=1 TO 10
2150 N=INT(52*RND+1)
2160 IF (N)26)*(N(33)THEN 2150
2170 IF U(N)=1 THEN 2150
2180 U(N)=1
2190 RESTORE 900
2200 FOR RD=1 TO N
2210 READ L$,CODE$
2220 NEXT RD
2230 IF RND).45 THEN Q$(I)=CODE$ :: A$(I)=SEG$(L$,1,1)ELSE Q$(I)=SEG$(L$,1,1)::
A$(I)=CODE$
2240 NEXT I
2250 CALL CLEAR
2260 FOR I=1 TO 10
2270 IF SEG$(Q$(I),1,1)="_" AND A$(I)="_._._." OR SEG$(Q$(I),1,1)="." AND A$(I)=
"._._._" THEN DISPLAY AT(I*2-1,1):"  CODE:" :: GOTO 2290
2280 IF SEG$(Q$(I),1,1)="." OR SEG$(Q$(I),1,1)="_" THEN DISPLAY AT(I*2-1,1):"DEC
ODE:" ELSE DISPLAY AT(I*2-1,1):"  CODE:"
2290 DISPLAY AT(I*2-1,9):Q$(I)
2300 ACCEPT AT(I*2-1,16)BEEP:AN$
2310 IF AN$<>A$(I)THEN CALL SOUND(110,220,2,330,3):: DISPLAY AT(I*2,1):"THE RIGH
T ANSWER IS ";A$(I)
2320 IF AN$=A$(I)THEN DISPLAY AT(I*2,1):"YOU ARE CORRECT!" :: RIGHT=RIGHT+1
2330 NEXT I
2340 SCORE=RIGHT*10 :: DISPLAY AT(22,4):"YOUR SCORE IS ";STR$(SCORE);"%"
2350 DISPLAY AT(24,2):"PRESS ANY KEY TO CONTINUE"
```

```
2360 CALL KEY(0,K,S):: IF S=0 THEN 2360
2370 SUBEND
2380 SUB SOUNDER(M,N)
2390 FOR I=1 TO 300 :: NEXT I
2400 DISPLAY AT(1,4):"NOW SOUNDING-OUT CODE" :: DISPLAY AT(24,1):""
2410 FOR I=M TO 24 STEP N :: FOR J=2 TO 31 :: CALL GCHAR(I,J,C)
2420 IF C=32 THEN CALL SOUND(250,42000,30):: X=X+1
2430 IF X=5 THEN 2470
2440 IF C=46 THEN CALL SOUND(150,2975,0):: CALL SOUND(150,42000,30):: X=0
2450 IF C=95 THEN CALL SOUND(300,2125,1):: CALL SOUND(150,42000,30):: X=0
2460 NEXT J :: NEXT I
2470 DISPLAY AT(1,1):"" :: DISPLAY AT(24,4):"SOUND-OFF COMPLETED" :: X=0
2480 SUBEND
2490 END
```

SPEED READER

Create your own text files that may be used to improve your reading speed with this TI Extended BASIC program. The text you enter may be saved on either cassette or disk for future use. If you select the CREATE TEXT FILE option from the main menu, you will be given a chance to EDIT or SAVE the text when you are finished entering it. Then, if you select the QUIT option in this mode, you are returned to the main menu so that you may create another file or load an existing one or read the text in memory.

When you are in the READ TEXT mode, you may set your own speed, from very slow (1) to very fast (50). A zero speed allows you to go at your own pace by pressing a key to display each line of text. The theory behind this type of tachistoscopic drill is that you will read more efficiently if your eye is trained to read entire phrases rather than individual words. You may want to enter text for beginners using short lines, perhaps centering the words on the line. Then you can work up to text which has been entered using the full 28 column line. Remember that if you select QUIT at the main menu, the program will end; you will have to run the program again and load a text file to do any more reading.

```
100 REM *SPEED READER* TI EXTENDED BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM A$(300):: R$=RPT$(" ",28)
140 CALL CLEAR :: DISPLAY AT(3,6):"= SPEED READER ="
150 L$=RPT$("=",28):: DISPLAY AT(5,1):L$ :: DISPLAY AT(8,3):"1 - LOAD TEXT FROM
CS1" :: DISPLAY AT(10,3):"2 - LOAD TEXT FROM DSK1"
160 DISPLAY AT(12,3):"3 - CREATE TEXT FILE" :: DISPLAY AT(14,3):"4 - READ TEXT"
170 DISPLAY AT(16,3):"5 - QUIT" :: DISPLAY AT(20,7):"YOUR CHOICE:"
180 DISPLAY AT(23,1):L$ :: ACCEPT AT(20,19)VALIDATE("12345")BEEP SIZE(1):CH
190 CALL CLEAR :: ON CH GOTO 200,250,300,570,710
200 OPEN #1:"CS1",INPUT ,FIXED 192,INTERNAL
210 FOR I=1 TO 300 STEP 6 :: INPUT #1:A$(I),A$(I+1),A$(I+2),A$(I+3),A$(I+4),A$(I
+5)
220 FOR J=I TO I+5 :: IF A$(J)="#" THEN 240
230 NEXT J :: NEXT I
240 CLOSE #1 :: GOTO 140
250 DISPLAY AT(10,6):"ENTER FILENAME:" :: DISPLAY AT(12,7):"DSK1."
```

```
260 ACCEPT AT(12,12)BEEP SIZE(10):F$ :: OPEN #1:"DSK1."&F$,INPUT ,FIXED 30,INTER
NAL
270 FOR I=1 TO 300 :: INPUT #1:A$(I):: IF EOF(1)THEN 290
280 NEXT I
290 CLOSE #1 :: GOTO 140
300 DISPLAY AT(1,1):"ENTER UP TO 300 LINES OF" :: DISPLAY AT(2,1):"TEXT, UP TO 2
8 CHARACTERS" :: DISPLAY AT(3,1):"PER LINE.  ENTER A NUMBER"
310 DISPLAY AT(4,1):"SIGN (#) ON A NEW LINE WHEN" :: DISPLAY AT(5,1):"YOU ARE FI
NISHED." :: DISPLAY AT(7,1):L$
320 FOR I=1 TO 300
330 DISPLAY AT(9,1):"LAST LINE=" :: DISPLAY AT(11,1):A$(I-1):: DISPLAY AT(13,1):
L$
340 DISPLAY AT(16,1):"ENTER LINE #";STR$(I);":" :: ACCEPT AT(20,1)SIZE(28)BEEP:A
$(I):: IF A$(I)="#" THEN 360
350 DISPLAY AT(20,1):"" :: NEXT I
360 CALL CLEAR :: DISPLAY AT(4,1):L$ :: DISPLAY AT(9,6):"1 - EDIT TEXT" :: DISPL
AY AT(11,6):"2 - SAVE ON CS1"
370 DISPLAY AT(13,6):"3 - SAVE ON DSK1" :: DISPLAY AT(15,6):"4 - QUIT" :: DISPLA
Y AT(19,7):"YOUR CHOICE:" :: DISPLAY AT(22,1):L$
380 ACCEPT AT(19,19)SIZE(1)BEEP VALIDATE("1234"):CH :: CALL CLEAR :: ON CH GOTO
390,480,530,140
390 DISPLAY AT(3,3):"THERE ARE ";STR$(I-1);" LINES" :: DISPLAY AT(6,3):"STARTING
 LINE #:1" :: ACCEPT AT(6,19)SIZE(-3)VALIDATE(DIGIT)BEEP:SL
400 DISPLAY AT(9,3):"ENDING LINE #:";STR$(I-1):: ACCEPT AT(9,17)SIZE(-3)VALIDATE
(DIGIT)BEEP:EL
410 FOR J=SL TO EL
420 CALL CLEAR :: DISPLAY AT(3,2):"PRESS E TO EDIT THIS LINE" :: DISPLAY AT(4,2)
:"PRESS ENTER TO CONTINUE" :: DISPLAY AT(7,1):L$
430 DISPLAY AT(1,10):"LINE #";STR$(J)
440 DISPLAY AT(12,1):A$(J)
450 CALL KEY(0,KEY,STATUS):: IF KEY=13 THEN 470 ELSE IF KEY<>69 OR STATUS=0 THEN
 450
460 DISPLAY AT(15,1):"RE-ENTER LINE:" :: ACCEPT AT(17,1)SIZE(28)BEEP:A$(J)
470 NEXT J :: GOTO 360
480 OPEN #1:"CS1",OUTPUT,INTERNAL,FIXED 192
490 FOR J=1 TO I STEP 6 :: PRINT #1:A$(J),A$(J+1),A$(J+2),A$(J+3),A$(J+4),A$(J+5
)
500 FOR K=J TO J+5 :: IF A$(K)="#" THEN 520
510 NEXT K :: NEXT J
520 CLOSE #1 :: GOTO 360
530 DISPLAY AT(10,6):"ENTER FILENAME:" :: DISPLAY AT(12,7):"DSK1."
540 ACCEPT AT(12,12)BEEP SIZE(10):F$ :: OPEN #1:"DSK1."&F$,OUTPUT,FIXED 30,INTER
NAL
550 FOR J=1 TO I :: PRINT #1:A$(J):: NEXT J
560 CLOSE #1 :: GOTO 360
570 DISPLAY AT(5,8):"SELECT SPEED:" :: DISPLAY AT(7,9):"1 = SLOWEST" :: DISPLAY
AT(9,8):"50 = FASTEST" :: DISPLAY AT(11,9):"0 = SELF-PACED"
580 DISPLAY AT(15,5):"YOUR CHOICE (0-50):" :: ACCEPT AT(15,24)BEEP SIZE(2)VALIDA
TE(DIGIT):SP
590 IF SP>0 THEN 650
600 CALL SOUND(100,1000,0):: DISPLAY AT(20,1):"PRESS A KEY TO DISPLAY EACH LINE
OF TEXT"
610 FOR J=1 TO 300
620 IF A$(J)="#" THEN 700
630 CALL KEY(0,KEY,STATUS):: IF STATUS=0 THEN 630
640 CALL CLEAR :: DISPLAY AT(12,1):A$(J):: FOR D=1 TO 5 :: NEXT D :: NEXT J :: G
OTO 140
650 CALL SOUND(100,1000,0):: DISPLAY AT(24,3):">PRESS ANY KEY TO BEGIN<"
660 CALL KEY(0,KEY,STATUS):: IF STATUS=0 THEN 660
670 FOR J=1 TO 300 :: IF A$(J)="#" THEN 700
680 CALL CLEAR :: DISPLAY AT(12,1):A$(J):: FOR D=1 TO 505-(10*SP):: NEXT D
690 NEXT J
700 CALL SOUND(100,1000,0):: PRINT TAB(13);"STOP" :: FOR D=1 TO 500 :: NEXT D ::
 GOTO 140
710 STOP
```

PLOT

Now you can use high-resolution pixel graphics in your TI BASIC programs without knowing anything about Assembly Language. This "PLOT" routine, by John Clulow, allows you to display single dots at any of 49,152 locations on the screen. Dot-addressable graphics let you do things like draw circles and lines, which are usually difficult in BASIC. Two versions are printed here, one for TI BASIC with the Mini-Memory module, the other for TI Extended BASIC with Memory Expansion. Both versions are designed to be added to your own programs, so they can be used whether you load from cassette or disk. An added feature is that the routine adds 16 user-definable characters to TI Extended BASIC. Two examples of how the routine can be applied, CIRCLE and SINE AND COSINE, are also listed.

When you want to access the PLOT routine in your program, use a statement with the following format:

CALL LINK("PLOT",dot-row,dot-column, starting-char-code,current-char)

The subprogram utilizes the standard character sets (ASCII codes 32-159) and redefines them as you specify. DOT-ROW and DOT-COLUMN are pixel locations with values of 0-191 and 0-255 respectively. Pixel number 0,0 would be the extreme upper left corner of the screen, while 191,255 should be the lower right dot. These parameters may be entered either as numbers or simple variables. (The routine does not support arrays.)

STARTING-CHAR-CODE determines the ASCII code of the first character to be used in pattern re-definition. It may be set to any value between 32 and 159; values outside that range will give a BAD VALUE error. Because this parameter also serves as a return variable for the highest character code used, it *must* be entered as a simple variable, not as a number or a subscripted variable.

CURRENT-CHAR is a return variable for the ASCII code of the character used in the most recent pattern re-definition. PLOT determines whether the character being displayed at the specified dot-row and dot-column has a lower ASCII code than that of the starting character. If so, the pattern of that character plus the new pixel are copied into the pattern of the next character to be defined. If not, the pixel is simply added to the pattern of the character presently being displayed. For example, if the dot-row and dot-column define a screen location occupied by the letter A (ASCII code 65), if the starting character is 96 and the current character 102, then the pixel will be added to the *pattern* for A and the new pattern placed in the next definable character code, 103. Character code 103, then, is displayed at the screen location formerly occupied by the letter A. Thus, pixels may be placed on the screen without appearing to alter the existing display.

On the other hand, if starting-character had been set at 45 in the above example, then the pixel would be added to the pattern for the letter A, but ASCII code 65 would continue to be displayed on the screen. This is because 65 is not less than 45. A BAD VALUE error will be reported if dot-row and dot-column are less than zero or greater than 9999. When either value exceeds the limit of the screen (191 for dot-row and 255 for dot-column), the placement of the pixel "wraps around" to the opposite side of the screen automatically.

Before the first CALL LINK statement in your program, character patterns for character codes beginning at 128 should be cleared:

```
100 FOR I=128 TO 159 (use 143 for Extended BASIC)
110 CALL CHAR(I,"")
120 NEXT I
```

If the Extended BASIC version of the subprogram is used (with Memory Expansion) and automatic sprite motion will not be used in the program, the routine allows you to re-define characters 144 to 159, which are normally not accessible in Extended BASIC. (Note that you cannot use the CALL CHAR or CALL COLOR statements to directly define these two character sets). With PLOT, if you specify a starting character value of 144 to 159, the subprogram will use that character code in starting pattern re-definition. You can also start with a code of 143 or less and work up to a character code of 144 to 159. The foreground and background colors of character sets 15 and 16 are automatically set to the same colors as character set 14 when the first LINK or PLOT is encountered in your program.

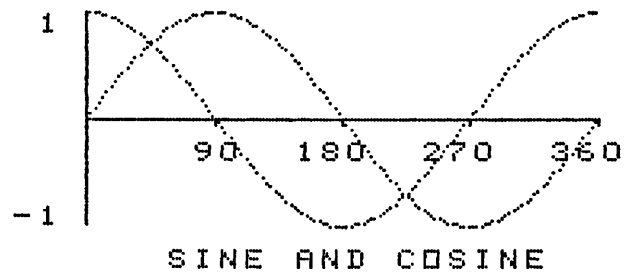
```
100 REM *PLOT* TI EXTENDED BASIC VERSION
110 REM BY JOHN CLULOW AND BERNIE ELSNER
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 REM 32K MEMORY EXPANSION REQUIRED, START AT MEMORY LOCATION >2700
150 DISPLAY AT(10,3)ERASE ALL:"LOADING MACHINE LANGUAGE"
160 CALL INIT :: CALL LOAD(-31878,0,"",8196,63,248,"",16376,80,76,79,84,32,32,39
,20):: MEM=9992
170 FOR I=1 TO 412 :: READ X :: CALL LOAD(MEM,X):: MEM=MEM+1 :: NEXT I
180 DATA 0,1,64,65,96,100,128,192,255,191,191,0,194,139,2,0,8,29,4,32,32,40,2,0,
8,30
190 DATA 4,32,32,32,2,0,8,31,4,32,32,32,4,192,2,1,0,3,6,160,40,42,176,160,39
200 DATA 12,152,2,39,14,26,9,152,2,39,18,19,6,112,160,39,9,216,2,39,18,216,2,39,
17
210 DATA 4,192,2,1,0,1,6,160,40,42,152,2,39,15,26,2,112,160,39,15,208,194,9,51,4
220 DATA 192,2,1,0,2,6,160,40,42,9,50,4,196,209,3,6,196,10,84,4,192,208,2,6,192
230 DATA 161,0,4,197,6,195,9,83,209,67,6,197,4,199,6,194,9,82,209,194,6,199,5,13
5,2
240 DATA 6,128,0,6,7,19,2,9,22,16,252,192,4,4,32,32,40,4,192,208,1,209,193,6,192
250 DATA 10,48,2,1,39,0,2,2,0,8,4,32,32,44,249,70,39,0,152,7,39,17,27,16,184
260 DATA 32,39,9,39,18,152,32,39,8,39,18,19,34,209,224,39,18,4,192,208,160,39,18
,2,1
270 DATA 0,3,6,160,40,100,4,192,208,7,6,192,10,48,2,1,39,0,2,2,0,8,4,32,32
280 DATA 36,4,192,2,1,0,4,208,135,6,160,40,100,192,4,208,71,4,32,32,32,194,202,4
,192
290 DATA 216,0,131,124,4,91,2,0,30,0,4,32,32,52,4,194,4,32,32,12,152,32,131,74,3
9
300 DATA 8,19,6,152,32,131,74,39,10,22,3,208,160,131,75,4,91,152,32,131,74,39,11
,22,234
```

```
310 DATA 4,192,208,32,131,75,6,192,176,160,39,13,6,0,22,252,176,160,131,76,4,91,
2,3,0
320 DATA 7,216,192,131,74,6,3,22,252,2,3,0,1,112,160,39,12,216,32,39,10,131,74,1
52,2
330 DATA 39,13,26,11,216,32,39,11,131,74,184,32,39,9,131,75,112,160,39,13,2,3,0,
2,16
340 DATA 242,216,194,131,74,4,32,32,8,4,91
350 REM BEGIN YOUR TI EXTENDED BASIC PROGRAM HERE...


100 REM *PLOT* TI BASIC/MINI-MEMORY VERSION
110 REM BY JOHN CLULOW AND BERNIE ELSNER
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 REM TI MINI-MEMORY MODULE REQUIRED, START AT MEMORY LOCATION >7D00
150 CALL CLEAR
160 PRINT TAB(3);"LOADING MACHINE LANGUAGE": : : : : : : : : : :
170 CALL INIT
180 CALL LOAD(28702,127,248,"",32760,80,76,79,84,32,32,125,20)
190 MEM=32008
200 FOR I=1 TO 388
210 READ X
220 CALL LOAD(MEM,X)
230 MEM=MEM+1
240 NEXT I
250 DATA 0,1,64,65,96,100,128,192,255,0,191,191,194,139,4,192,2,1,0,3,6,160,126,
18,176
260 DATA 160,125,12,152,2,125,14,26,9,152,2,125,19,19,6,112,160,125,9,216,2,125,
19,216,2
270 DATA 125,18,4,192,2,1,0,1,6,160,126,18,152,2,125,15,26,2,112,160,125,15,208,
194,9
280 DATA 51,4,192,2,1,0,2,6,160,126,18,9,50,4,196,209,3,6,196,10,84,4,192,208,2
290 DATA 6,192,161,0,4,197,6,195,9,83,209,67,6,197,4,199,6,194,9,82,209,194,6,19
9,5
300 DATA 135,2,6,128,0,6,7,19,2,9,22,16,252,192,4,4,32,96,44,4,192,208,1,209,193
310 DATA 6,192,10,48,2,1,125,0,2,2,0,8,4,32,96,48,249,70,125,0,152,7,125,18,27
320 DATA 16,184,32,125,9,125,19,152,32,125,8,125,19,19,34,209,224,125,19,4,192,2
08,160,125,19
330 DATA 2,1,0,3,6,160,126,76,4,192,208,7,6,192,10,48,2,1,125,0,2,2,0,8,4
340 DATA 32,96,40,4,192,2,1,0,4,208,135,6,160,126,76,192,4,208,71,4,32,96,36,194
,202
350 DATA 4,192,216,0,131,124,4,91,2,0,19,0,4,32,96,80,4,194,4,32,96,68,152,32,13
1
360 DATA 74,125,8,19,6,152,32,131,74,125,10,22,3,208,160,131,75,4,91,152,32,131,
74,125,11
370 DATA 22,234,4,192,208,32,131,75,6,192,176,160,125,13,6,0,22,252,176,160,131,
76,4,91,2
380 DATA 3,0,7,216,192,131,74,6,3,22,252,2,3,0,1,112,160,125,12,216,32,125,10,13
1,74
390 DATA 152,2,125,13,26,11,216,32,125,11,131,74,184,32,125,9,131,75,112,160,125
,13,2,3,0
400 DATA 2,16,242,216,194,131,74,4,32,96,64,4,91
410 REM ONCE THIS PROGRAM IS LOADED INTO MINI-MEMORY IT NEED NOT BE RELOADED
420 REM UNLESS CALL INIT HAS BEEN USED OR PROGRAM IS ALTERED
```

SINE AND COSINE

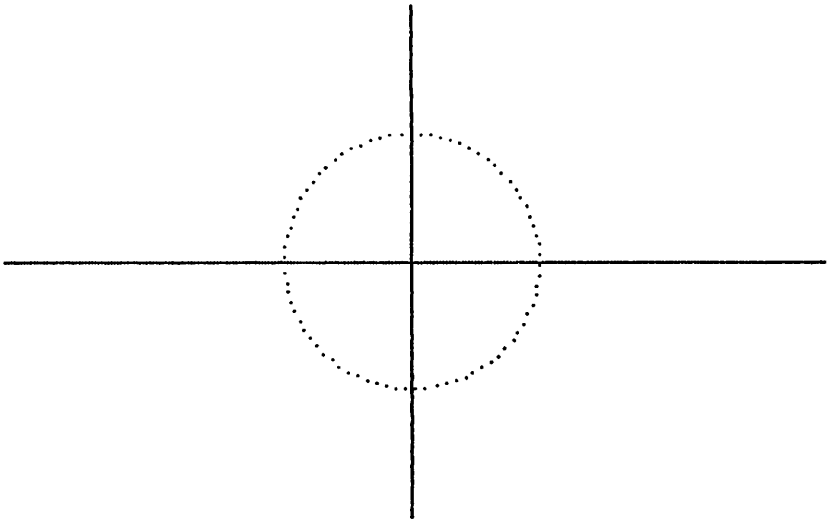
```
500 REM  *SINE & COSINE* FOR USE WITH PLOT
510 REM  TI BASIC & MINI-MEMORY OR
520 REM  TI EXTENDED BASIC & MEMORY EXP.
530 REM  BY JOHN CLULOW AND BERNIE ELSNER
540 REM  FROM PROGRAMS FOR THE TI HOME COMPUTER
550 REM   COPYRIGHT (C) 1983 BY STEVE DAVIS
560 CALL CLEAR
570 PRINT TAB(8);"SINE AND COSINE":::::
580 REM  FOR EXTENDED BASIC CHANGE 159 TO 143
590 FOR I=96 TO 159
600 CALL CHAR(I,"")
610 NEXT I
620 REM  DRAW X AND Y AXIS, START AT CHAR 84
630 S=84
640 FOR I=48 TO 55
650 CALL LINK("PLOT",96,I,S,C)
660 CALL LINK("PLOT",I+8,47,S,C)
670 NEXT I
680 CALL HCHAR(13,8,C-1,19)
690 CALL VCHAR(9,6,C,9)
700 REM  LABELS AND 90 DEG MARKS
710 REM  ADVANCE START CHARACTER
720 S=S+1
730 LABEL$="90  180  270  360"
740 FOR I=1 TO LEN(LABEL$)
750 CALL HCHAR(14,I+10,ASC(SEG$(LABEL$,I,1)))
760 NEXT I
770 CALL HCHAR(8,5,49)
780 CALL HCHAR(17,4,45)
790 CALL HCHAR(17,5,49)
800 FOR I=87 TO 208 STEP 40
810 CALL LINK("PLOT",97,I,S,C)
820 CALL LINK("PLOT",98,I,S,C)
830 NEXT I
840 REM   PLOT SINE/COSINE
850 PI=3.14159265359
860 FOR RADIANS=0 TO 2*PI STEP .05
870 DOTROW=96-INT(40*SIN(RADIANS)+.5)
880 DOTCOL=47+INT(25.46479*RADIANS+.5)
890 CALL LINK("PLOT",DOTROW,DOTCOL,S,C)
900 DOTROW=96-INT(40*COS(RADIANS)+.5)
910 CALL LINK("PLOT",DOTROW,DOTCOL,S,C)
920 NEXT RADIANS
930 CALL KEY(0,K,S)
940 IF S=0 THEN 930
950 END
```



SINE AND COSINE

CIRCLE

```
500 REM  *CIRCLE* FOR USE WITH PLOT
510 REM  TI BASIC & MINI-MEMORY OR
520 REM  TI EXTENDED BASIC & MEMORY EXP.
530 REM  BY JOHN CLULOW AND BERNIE ELSNER
540 REM  FROM PROGRAMS FOR THE TI HOME COMPUTER
550 REM   COPYRIGHT (C) 1983 BY STEVE DAVIS
560 CALL CLEAR
570 REM  FOR EXTENDED BASIC CHANGE 159 TO 143
580 FOR I=96 TO 159
590 CALL CHAR(I,"")
600 NEXT I
610 REM  DRAW X AND Y AXIS, START AT CHAR 93
620 S=93
630 FOR I=0 TO 7
640 CALL LINK("PLOT",96,I,S,C)
650 CALL LINK("PLOT",I,128,S,C)
660 NEXT I
670 CALL HCHAR(13,2,C-1,31)
680 CALL VCHAR(2,17,C,23)
690 REM  PATCH IN X AXIS, CHANGE STARTING CHAR
700 S=S+1
710 FOR I=128 TO 135
720 CALL LINK("PLOT",96,I,S,C)
730 NEXT I
740 REM  DRAW DOTTED LINE CIRCLE WITH A
750 REM  RADIUS OF 30, EXPAND Y BY 1.18
760 F=1.18
770 PI=3.14159265359
780 FOR RADIANS=0 TO 2*PI STEP .08
790 DOTROW=96-INT(F*40*SIN(RADIANS)+.5)
800 DOTCOL=128+INT(40*COS(RADIANS)+.5)
810 CALL LINK("PLOT",DOTROW,DOTCOL,S,C)
820 NEXT RADIANS
830 CALL KEY(0,K,S)
840 IF S=0 THEN 830
850 END
```



CRAWL

If you have ever noticed the messages that move across the bottom of your television screen when the station wants to bring an important announcement to your attention, you might have wondered how you could use such a device in your programs. This short routine in TI Extended BASIC allows you to have a message "crawl" across the screen until a key is pressed. To alter the speed of the crawl or change the message, just assign values to the variables D and C$. To make the crawl appear at the top or middle of the screen, change the row value in the Display At statement.

```
100 REM *CRAWL* TI EXTENDED BASIC
110 REM BY DAVID MIGICOVSKY
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 D=5 :: C$="THIS MESSAGE WILL CRAWL ACROSS THE BOTTOM OF THE SCREEN UNTIL A K
EY IS PRESSED."
150 C$=RPT$(" ",28)&C$ :: CALL CLEAR
160 FOR X=1 TO LEN(C$):: DISPLAY AT(24,1):SEG$(C$,X,28):: FOR Y=0 TO D :: NEXT Y
 :: CALL KEY(0,K,S):: IF S<>0 THEN 180
170 NEXT X :: GOTO 160
180 STOP
```

DECI-HEX CONVERTER

More and more people are using Assembly Language to program their home computers now because of the speed of program execution and variety of functions it offers. However, this often requires entering numbers or reading numbers in Hexidecimal. The numbering system we humans normally use is called Decimal, or Base 10. Computers think in Binary, or Base 2, but a system that is more human-readable than Base 2 but easily converted to Base 2 is Hexidecimal, which is a Base 16 numbering system.

Since our Base 10 system only has ten symbols for digits (0 through 9), the first six letters of our alphabet are used for digits ten through fifteen in Hexidecimal. For example, the number 13 in decimal stands for a value of one ten plus three ones. However, a number 13 in Hex would stand for a value of one sixteen plus three ones, or 19 in decimal. The number A in Hex is the same as 10 in decimal. Thus, the number A5 in Hex would be 165 in decimal (10 sixteens plus 5 ones).

As you might imagine, if you are not used to doing such conversions, they can be a tedious and time-consuming process that can result in some confusion and error on the part of the programmer.

This program very simply accepts a decimal number as input then prints out the Hexidecimal number of the same value. Optionally, you may use a printer so you will have a copy of the conversions to refer to, or you can print a chart of a whole list of numbers converted from decimal to Hex.

```
100 REM *DECIMAL TO HEX NUMBER CONVERTER* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CLEAR
140 INPUT "USING A PRINTER? (Y/N)":Y$
150 IF (Y$<>"Y")*(Y$<>"y")THEN 190
160 INPUT "PRINTER DEVICE NAME:":DN$
170 OPEN #1:DN$,OUTPUT
180 PRINT #1:"DECIMAL    HEX": :
190 H=16
200 CALL CLEAR
210 PRINT "ENTER THE DECIMAL NUMBER YOU":"WANT TO CONVERT:"
220 INPUT N
230 A$=""
240 NP$=STR$(N)
250 J=1
260 FOR I=1 TO 10
270 K(I)=J*H
280 J=K(I)
290 NEXT I
300 FOR I=10 TO 1 STEP -1
310 IF K(I)<=N THEN 340
320 L(I)=0
330 GOTO 360
340 L(I)=INT(N/K(I))
350 N=N-(L(I)*K(I))
360 NEXT I
370 L(0)=N
380 FOR I=10 TO 0 STEP -1
390 IF L(I)<>15 THEN 410
400 A$=A$&"F"
410 IF L(I)<>14 THEN 430
420 A$=A$&"E"
430 IF L(I)<>13 THEN 450
440 A$=A$&"D"
450 IF L(I)<>12 THEN 470
460 A$=A$&"C"
470 IF L(I)<>11 THEN 490
480 A$=A$&"B"
490 IF L(I)<>10 THEN 510
500 A$=A$&"A"
510 IF L(I)>=10 THEN 530
520 A$=A$&STR$(L(I))
530 NEXT I
540 PRINT A$
550 IF (Y$<>"Y")*(Y$<>"y")THEN 570
560 PRINT #1:NP$;TAB(12);A$
570 PRINT : : :"ANOTHER NUMBER? (Y/N)"
580 CALL KEY(0,KEY,STATUS)
590 IF STATUS=0 THEN 580
600 IF (KEY=78)+(KEY=110)THEN 630
610 IF (KEY=89)+(KEY=121)THEN 200
620 GOTO 580
630 STOP
```

HEXI-DEC CONVERTER

The reverse of the Deci-Hex program, this one accepts a Hexidecimal number (Base 16) and converts it to Decimal (Base 10).

If you have a printer, you can save your conversions for future reference. In converting Hex digits A through F, this program differs somewhat from similar conversion programs in that it uses the ASC function of TI BASIC to return the ASCII character code of each digit in the string, saving several IF-THEN steps and simplifying error checking.

```
100 REM *HEX TO DECIMAL NUMBER CONVERTER* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 CALL CLEAR
140 PRINT "HEX TO DECIMAL CONVERSION": :"WILL YOU USE A PRINTER?":"(Y OR N)"
150 CALL SOUND(100,1400,3)
160 CALL KEY(0,PKEY,STATUS)
170 IF STATUS=0 THEN 160
180 IF PKEY()89 THEN 220
190 INPUT "DEVICE NAME:":DEV$
200 OPEN #1:DEV$,OUTPUT
210 PRINT #1:"HEX";TAB(12);"DECIMAL": :
220 PRINT
230 INPUT "HEX #: ":H$
240 N=0
250 P=0
260 FOR I=LEN(H$)TO 1 STEP -1
270 D$=SEG$(H$,I,1)
280 IF ASC(D$)(58 THEN 320
290 IF ASC(D$)(65 THEN 340
300 D=ASC(D$)-55
310 GOTO 330
320 D=VAL(D$)
330 IF (D(16)*(D)=0)THEN 370
340 PRINT "INVALID INPUT":"HEX DIGITS MUST BE 0-F"
350 CALL SOUND(150,120,0)
360 GOTO 230
370 N=N+D*(16^P)
380 P=P+1
390 NEXT I
400 PRINT "DECIMAL #: ";N
410 IF PKEY()89 THEN 430
420 PRINT #1:H$;TAB(12);N
430 PRINT
440 PRINT "PRESS Q TO QUIT OR":"ANY OTHER KEY TO CONTINUE"
450 CALL KEY(0,KEY,STATUS)
460 IF STATUS=0 THEN 450
470 IF KEY()81 THEN 220
480 STOP
```

LOWER CASE

The lower case character set (ASCII codes 97-122) on the 99/4A appears on your monitor as small capitals. This routine can be included in any TI BASIC or TI Extended BASIC program to define this character set as "true lower case" letters. Though it is not easy to have "descenders" (the tails of the j, g, q, p and y) below the line without messing up your video display format, now you can at least use lower case in your screen text and have it look like lower case. To save a few bytes of memory, letters not using the top three rows of dots in the character block do not have the leading zeroes of the Hex code in the Data statements. This is justified in the Read loop by putting the zeroes back in where they belong. Lines 150 to 190 are just a demonstration that allow you to type in some text to see what the letters look like. The backslash (character 92) has been redefined as a small picture of TI's home state. When you get the Texas prompt, just type in your lower case text. To stop, press Clear. To use the routine in a program, of course, you would want to eliminate the demonstration section.

```
100 REM * Lower Case Letters * TI BASIC
110 REM BY DAVID MIGICOVSKY
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 REM DEMONSTRATION
150 GOSUB 210
160 CALL CLEAR
170 CALL CHAR(92,"30303FFFFE7C180C")
180 INPUT "\)":N$
190 GOTO 180
200 REM LOWER CASE ROUTINE
210 FOR X=1 TO 26
220 CALL CHAR(X+96,"")
230 READ F$
240 IF SEG$(F$,1,1)="0" THEN 260
250 F$="000000"&F$
260 CALL CHAR(X+96,F$)
270 NEXT X
280 RETURN
290 DATA 3848484834,0060203824242478,3844404438,000C08384848483C,38447C4038,0018
24207020202,18241C0438
300 DATA 0060202834242424,0010007010101 07C,0008001808084 83,0020202428302824,0030
10101010107C,A854545454
310 DATA 5824242424,3844444438,782438207,304838080C,582420202,3C40380478,0020207
820202418,4848484834
320 DATA 444428281,6A2A2A2A14,4428102844,4428101010,7C4810247C
```

SCREEN SPEAK

Those who have used the Terminal Emulator II with the Speech Synthesizer for telecommunications uses are probably familiar with the "Speak" function. With this feature, when you press Control 1, the computer speaks the text that is currently displayed on the screen. This TI BASIC routine allows you to add this function to any of your TI BASIC programs. When a CALL KEY statement is used in your program, test the value of the return variable (ASCII code of the key

pressed). If it is 177 (the code for Control 1) then you can go to the routine that reads the data on the screen and sends it a line at a time to the Speech Synthesizer. If another key is more convenient, you merely need to specify the code for that key instead of 177.

```
100 REM *SCREEN SPEAK* TI BASIC
110 REM SPEECH SYNTHESIZER AND TERMINAL EMULATOR 2 REQUIRED
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 OPEN #1:"SPEECH",OUTPUT
150 CALL CLEAR
160 PRINT "THIS IS A DEMONSTRATION OF":"HOW THE COMPUTER WILL":"SPEAK TEXT THAT
IS"
170 PRINT "DISPLAYED ON THE SCREEN.":"PRESS CONTROL 1 TO HEAR":"ME TALK.  WHEN I
 AM":"FINISHED YOU MAY PRESS"
180 PRINT "CLEAR TO STOP.":::::::::::::::
190 CALL KEY(0,K,S)
200 IF S=0 THEN 190
210 IF K<>177 THEN 190
220 FOR R=1 TO 24
230 A$=""
240 FOR C=3 TO 30
250 CALL GCHAR(R,C,X)
260 A$=A$&CHR$(X)
270 NEXT C
280 PRINT #1:A$
290 NEXT R
300 GOTO 190
```

SPEECH HELPER

If you are writing programs that will contain text-to-speech using the Terminal Emulator II (or the TI Text-To-Speech disk) and the TI Speech Synthesizer, this menu-driven program should make things a little easier. You can enter a phrase in English (with optional pitch setting) to hear how it would be spoken in your program. Or, you can have the computer print a list of the constituent allophones in a phrase. If a phrase does not sound exactly right, you can then look for an allophone that would be more appropriate (in the list in the TE2 manual). Also, you can enter allophone numbers and hear how the computer would speak them.

```
100 REM   *SPEECH HELPER* TI BASIC
110 REM  FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM  COPYRIGHT (C) 1983 BY STEVE DAVIS
130 REM   TERMINAL EMULATOR 2 & SPEECH SYNTHESIZER REQUIRED; PRINTER OPTIONAL
140 CALL CLEAR
150 PRINT " TE2 SPEECH - CHOOSE ONE:"
160 PRINT ::" 1-ENTER PHRASE, COMPUTER":"   SPEAKS"
170 PRINT :" 2-ENTER PHRASE, COMPUTER":"   PRINTS ALLOPHONE NUMBERS"
180 PRINT :" 3-ENTER ALLOPHONE NUMBERS,":"   COMPUTER SPEAKS WORD"
190 PRINT :" 4-END PROGRAM"::
200 INPUT SEL
210 IF (SEL<1)+(SEL>4)THEN 200
220 CALL CLEAR
230 ON SEL GOTO 240,350,620,820
240 OPEN #1:"SPEECH",OUTPUT
250 PRINT "ENTER A PHRASE TO BE SPOKEN."
```

```
260 PRINT "TO ALTER PITCH, ENTER //":"FOLLOWED BY PITCH # (0-63)":"AND SLOPE # (
0-255).":"EXAMPLE: //43 128"::
270 PRINT "(ENTER 999 TO EXIT)"
280 PRINT
290 INPUT "PHRASE=":A$
300 IF A$="999" THEN 330
310 PRINT #1:A$
320 GOTO 290
330 CLOSE #1
340 GOTO 140
350 OPEN #1:"SPEECH",OUTPUT
360 OPEN #2:"ALPHON",INTERNAL
370 PRINT "ENTER A WORD OR PHRASE.":"COMPUTER WILL PRINT THE":"ALLOPHONE NUMBERS
.":
380 INPUT "USING A PRINTER? (Y/N)":Y$
390 IF Y$<>"Y" THEN 420
400 INPUT "DEVICE NAME:":DN$
410 OPEN #3:DN$,OUTPUT
420 PRINT :"(ENTER 999 TO QUIT)"::
430 INPUT "PHRASE=":A$
440 IF A$="" THEN 430
450 IF A$="999" THEN 570
460 PRINT #1:A$
470 INPUT #2:B$
480 P$=""
490 Z=LEN(B$)
500 FOR R=4 TO Z
510 PRINT ASC(SEG$(B$,R,1))
520 P$=P$&STR$(ASC(SEG$(B$,R,1)))&"  "
530 NEXT R
540 IF Y$<>"Y" THEN 560
550 PRINT #3::A$:P$
560 GOTO 420
570 CLOSE #1
580 CLOSE #2
590 IF Y$<>"Y" THEN 140
600 CLOSE #3
610 GOTO 140
620 OPEN #1:"ALPHON",INTERNAL
630 PRINT "ENTER NUMBER OF SYLLABLES INTHE WORD THEN ENTER EACH    ALLOPHONE NUM
BER 1 AT A TIME"
640 PRINT :"(ENTER 999 TO QUIT)"::
650 INPUT "SYLLABLES=":A
660 IF A=999 THEN 800
670 B$=CHR$(250)&CHR$(255)&CHR$(A)
680 PRINT :"(ENTER 0 TO END WORD)"::
690 INPUT "ALLOPHONE NUMBER=":C
700 IF C=0 THEN 730
710 B$=B$&CHR$(C)
720 GOTO 690
730 PRINT #1:B$
740 PRINT :"(PRESS ENTER FOR NEW WORD OR ANOTHER KEY TO REPEAT)"
750 CALL KEY(0,KEY,STATUS)
760 IF STATUS=0 THEN 750
770 IF KEY=13 THEN 640
780 PRINT #1:B$
790 GOTO 750
800 CLOSE #1
810 GOTO 140
820 STOP
```

DISK LISTER

If you have several diskettes full of programs, you know that sometimes it is difficult to remember exactly where a program is when you need it. Also, as your diskette library grows, it becomes more and more bothersome to read through the catalog for each disk when searching for that elusive listing. One solution to that problem would be to use a data management program, such as TI's Personal Record Keeping module or a similar program you may have written yourself. But why bother typing in all those program and disk names when the computer has already recorded them on disk for you?

This program reads the catalog off of each diskette and stores it in the computer's RAM (Random Access Memory for you beginners). When you have finished loading the diskette catalogs, the program then sorts the programs in alphabetical order and prints the results on your printer. The name of the disk on which each program lies is printed beside each file name. You may also save the sorted list on disk so that you may add to it later.

```
100 REM *DISK LISTER* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM B$(300)
140 I=0
150 TB$="..............."
160 DEF T$=SEG$(TB$,1,14-LEN(C$))
170 CALL CLEAR
180 DISPLAY "ENTER":"1 - TO CREATE CATALOG FILE":"    FROM DISKS":"2 - TO LOAD E
XISTING FILE":"    FROM DISK"
190 INPUT CH
200 IF CH=2 THEN 860
210 IF CH<>1 THEN 190
220 DISPLAY "INSERT DISK TO READ, PRESS  ENTER"
230 INPUT Y$
240 OPEN #1:"DSK1.",INPUT ,RELATIVE,INTERNAL
250 INPUT #1:A$,Z,Z,Z
260 DISPLAY A$
270 I=I+1
280 INPUT #1:C$,Z,Z,Z
290 IF LEN(C$)=0 THEN 330
300 B$(I)=C$&T$&A$
310 DISPLAY B$(I)
320 GOTO 270
330 I=I-1
340 CLOSE #1
350 CALL SOUND(100,1000,0)
360 DISPLAY :")TO LOAD MORE DATA":" INSERT NEXT DISK AND":" PRESS C TO CONTINUE"
370 DISPLAY :")TO QUIT LOADING, PRESS Q"
380 CALL KEY(0,K,S)
390 IF K=67 THEN 240
400 IF (S=0)+(K<>81)THEN 380
410 CALL SOUND(100,1000,0)
420 DISPLAY :")SORTING..."
430 X=1
440 X=2*X
450 IF X<=I THEN 440
460 X=INT(X/2)
470 IF X=0 THEN 590
```

```
480 FOR J=1 TO I-X
490 Y=J
500 Z=Y+X
510 IF B$(Y)<=B$(Z)THEN 570
520 Q$=B$(Y)
530 B$(Y)=B$(Z)
540 B$(Z)=Q$
550 Y=Y-X
560 IF Y>0 THEN 500
570 NEXT J
580 GOTO 460
590 DISPLAY :"ENTER P TO PRINT":"   OR S TO SAVE ON DISK"
600 INPUT S$
610 IF S$="S" THEN 770
620 IF S$<>"P" THEN 600
630 INPUT "PRINTER DEVICE NAME:":DN$
640 OPEN #2:DN$,OUTPUT
650 FOR J=1 TO I
660 PRINT #2:B$(J)
670 NEXT J
680 CLOSE #2
690 DISPLAY :"ENTER Q TO QUIT ":"      P TO PRINT AGAIN":"      S TO SAVE ON DIS
K"
700 INPUT Y$
710 IF Y$="Q" THEN 760
720 IF Y$="S" THEN 770
730 IF Y$<>"P" THEN 700
740 INPUT "SAME DEVICE NAME? (Y/N)":Y$
750 IF Y$="Y" THEN 640 ELSE 630
760 STOP
770 INPUT "FILENAME:":FN$
780 IF SEG$(FN$,1,5)="DSK1." THEN 800
790 FN$="DSK1."&FN$
800 OPEN #3:FN$,OUTPUT,INTERNAL,FIXED 25
810 FOR J=1 TO I
820 PRINT #3:B$(J)
830 NEXT J
840 CLOSE #3
850 GOTO 950
860 INPUT "FILENAME:":FN$
870 IF SEG$(FN$,1,5)="DSK1." THEN 890
880 FN$="DSK1."&FN$
890 OPEN #1:FN$,INPUT ,INTERNAL,FIXED 25
900 I=0
910 I=I+1
920 INPUT #1:B$(I)
930 IF EOF(1)THEN 940 ELSE 910
940 CLOSE #1
950 DISPLAY :"ENTER A TO ADD TO FILE":"      P TO PRINT FILE":"      Q TO QUIT"
960 INPUT P$
970 IF P$="P" THEN 630
980 IF P$="Q" THEN 1000
990 IF P$<>"A" THEN 960 ELSE 220
1000 STOP
```

DISK LISTER DISPLAY

This program allows you to search for a disk by program name using the file created by the Disk Lister program. Also, those without printers may use this program to display the file on the screen.

```
100 REM *DISK LISTER DISPLAY* TI BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 DIM B$(300)
140 CALL CLEAR
150 PRINT "ENTER FILENAME:"
160 INPUT "DSK1.":FN$
170 DISPLAY "LOADING FILE..."
180 OPEN #1:"DSK1."&FN$,INPUT ,INTERNAL,FIXED 25
190 I=I+1
200 INPUT #1:B$(I)
210 IF EOF(1)THEN 220 ELSE 190
220 CLOSE #1
230 CALL CLEAR
240 DISPLAY "ENTER 1 TO DISPLAY FILE":"      2 TO SEARCH FILE BY           KEYWO
RD":"      3 TO QUIT"
250 INPUT CH
260 IF CH=2 THEN 440
270 IF CH=3 THEN 540
280 IF CH<>1 THEN 250
290 Q=0
300 CALL CLEAR
310 FOR J=1 TO I
320 DISPLAY B$(J)
330 Q=Q+1
340 IF (Q<22)*(J<>I)THEN 420
350 PRINT "    (C=CONTINUE  S=STOP)"
360 CALL KEY(0,KEY,STATUS)
370 IF STATUS=0 THEN 360
380 IF KEY=83 THEN 230
390 IF KEY<>67 THEN 360
400 Q=0
410 CALL CLEAR
420 NEXT J
430 GOTO 230
440 CALL CLEAR
450 INPUT "KEYWORD=":K$
460 DISPLAY "SEARCHING...": :
470 FOR J=1 TO I
480 IF POS(SEG$(B$(J),1,10),K$,1)=0 THEN 500
490 DISPLAY B$(J)
500 NEXT J
510 PRINT :" (PRESS A KEY TO CONTINUE)"
520 CALL KEY(0,KEY,STATUS)
530 IF STATUS=0 THEN 520 ELSE 230
540 STOP
```

TI-TLES

It is easy to create bold colored title letters with this TI Extended BASIC program. All you have to do is enter up to 4 lines of text, with up to 15 upper-case alphabetic characters per line, and choose the colors for the titles and the screen. The computer automatically centers the words on the screen as it draws them. Your titles remain on the screen until you press a key. If you wish to redraw the same titles, use the Delete key to move the words back to the left margin so that the computer can automatically center them again. (Or, you could use the Erase key and re-type the titles).

You will find many uses for bold letters in your display. If you own a video recorder, add titles to your home movies by plugging your computer's video cable or modulator into your recorder and recording the graphics you have created. Also, as with the Color Bar Graphs program, you could photograph the titles from your monitor to make title slides for your audio-visual presentations. (Try slide film at 1/30 second shutter speed. Experiment with exposure settings.)

Another use, of course, would be to use these title graphics in your own programs. To do this, change the values of L1$, L2$, L3$ and L4$ in line 150 to the words you want to display. Set C and C1 to the color number that you desire for the titles and screen respectively. Delete lines 160, 180-270 and 320-340. The line number reference at the end of the program would need to GOTO the next line to be executed in your program. Alternatively, you could use just the CALL CHAR statements from this listing that correspond with the letters you will need to draw in your program and use your own CALL HCHAR statements to draw the titles. To end this program, you may press the Clear key.

```
100 REM *TI-TLES* TI EXTENDED BASIC
110 REM BY BRIAN MADIGAN & DAVID MIGICOVSKY
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 REM FOR AUTOMATIC CENTERING, ENTER TITLES AT LEFT MARGIN. PRESS CLEAR TO END
 PROGRAM.
150 F$=RPT$(" ",15)!! L1$="TYPE IN YOUR" !! L2$="OWN TITLES HERE" !! L3$="OR PRE
SS ENTER" !! L4$="TO SEE THESE" !! C=2 !! C1=16
160 CALL CLEAR !! CALL SCREEN(16)!! FOR X=2 TO 7 !! CALL COLOR(X,5,1)!! NEXT X !
! DISPLAY AT(5,10):"TI-TLES" !! FOR X=1 TO 150 !! NEXT X
170 DIM L(4,15),A(26)!! L1=1 !! FOR Z=36 TO 136 STEP 4 !! A(L1)=Z !! L1=L1+1 !!
NEXT Z
180 CALL SCREEN(16)!! FOR X=0 TO 8 !! CALL COLOR(X,5,1)!! NEXT X !! CALL CLEAR
190 DISPLAY AT(1,2):"LINE 1:";L1$ !! DISPLAY AT(2,2):"LINE 2:";L2$ !! DISPLAY AT
(3,2):"LINE 3:";L3$ !! DISPLAY AT(4,2):"LINE 4:";L4$
200 CALL CHAR(140,"FFFFFFFFFFFFFF00FF8181818181FF")
210 DISPLAY AT(6,2):"" !! DISPLAY AT(7,2):"" !! ACCEPT AT(1,9)VALIDATE(UALPHA)SI
ZE(-15):L1$ !! ACCEPT AT(2,9)VALIDATE(UALPHA)SIZE(-15):L2$
220 ACCEPT AT(3,9)VALIDATE(UALPHA)SIZE(-15):L3$ !! ACCEPT AT(4,9)VALIDATE(UALPHA
)SIZE(-15):L4$
230 DISPLAY AT(6,2):"PRESS REDO TO CHANGE OR" !! DISPLAY AT(7,2):"PRESS PROC'D T
O CONTINUE"
240 CALL KEY(0,K,S)!! IF S=0 THEN 240 !! IF K=6 THEN 210
250 IF K<>12 THEN 240
```

```
260 DISPLAY AT(6,2):"" :: DISPLAY AT(7,2):"" :: FOR X=2 TO 16 :: CALL SPRITE(#X,
140,X,(X*8)+42,27):: NEXT X
270 CALL PATTERN(#16,141):: CALL COLOR(#16,2):: FOR X=2 TO 16 :: DISPLAY AT(X+6,
3):X :: NEXT X
280 Z=LEN(L1$):: Z=15-Z :: Z1=INT(Z/2):: Z=15-(Z1+LEN(L1$)):: L1$=SEG$(F$,1,Z)&L
1$&SEG$(F$,1,Z1)
290 Z=LEN(L2$):: Z=15-Z :: Z1=INT(Z/2):: Z=15-(Z1+LEN(L2$)):: L2$=SEG$(F$,1,Z)&L
2$&SEG$(F$,1,Z1)
300 Z=LEN(L3$):: Z=15-Z :: Z1=INT(Z/2):: Z=15-(Z1+LEN(L3$)):: L3$=SEG$(F$,1,Z)&L
3$&SEG$(F$,1,Z1)
310 Z=LEN(L4$):: Z=15-Z :: Z1=INT(Z/2):: Z=15-(Z1+LEN(L4$)):: L4$=SEG$(F$,1,Z)&L
4$&SEG$(F$,1,Z1)
320 ON WARNING NEXT :: DISPLAY AT(10,10):"CHOOSE A COLOR" :: DISPLAY AT(11,10):"
FOR THE TITLES:";C :: ACCEPT AT(11,26)SIZE(-3):C
330 DISPLAY AT(13,10):"FOR THE SCREEN:";C1 :: ACCEPT AT(13,26)SIZE(-3):C1
340 IF C)16 THEN 320 ELSE IF C(2 THEN 320 ELSE IF C1)16 THEN 330 ELSE IF C1(2 TH
EN 330
350 CALL DELSPRITE(ALL):: CALL CLEAR :: CALL SCREEN(C1):: FOR X=1 TO 14 :: CALL
COLOR(X,C,1):: NEXT X
360 L5$=L1$&L2$&L3$&L4$ :: Z1=1 :: FOR I=1 TO 4 :: FOR J=1 TO 15 :: IF SEG$(L5$,
Z1,1)="" THEN 380
370 Z=POS("ABCDEFGHIJKLMNOPQRSTUVWXYZ",SEG$(L5$,Z1,1),1):: L(I,J)=A(Z)
380 Z1=Z1+1 :: NEXT J :: NEXT I
390 CALL CHAR(36,"0F3F7F7F7C7878787F7F7F7F78787878F0FCFEFE3E1E1E1EFEFEFEFE1E1E1E
1E")
400 CALL CHAR(40,"7F7F7F7F78787F7F7F7F78787F7F7F7FF8FCFEFE1E1EFEF8F8FE1E1EFEFEFC
F8")
410 CALL CHAR(44,"0F3F7F7F7C7878787878787C7F7F3F0FFCFCFCFC000000000000000FCFCFC
FC")
420 CALL CHAR(48,"7F7F7F7F7878787878787878787F7F7F7FF0FCFEFE3E1E1E1E1E1E1E3EFEFEFC
F")
430 CALL CHAR(52,"7F7F7F7F78787F7F7F7F78787F7F7F7FFCFCFCFC0000E0E0E0E00000FCFCFC
FC")
440 CALL CHAR(56,"7F7F7F7F78787F7F7F7F787878787878FCFCFCFC0000E0E0E0E")
450 CALL CHAR(60,"0F3F7F7F7C7878787878787C7F7F3F0FFEFEFEFE0000FEFEFEFE1E1EFEFEFE
FE")
460 CALL CHAR(64,"7878787878787F7F7F7F7878787878781E1E1E1E1E1EFEFEFEFE1E1E1E1E1E
1E")
470 CALL CHAR(68,"1F1F1F1F0303030303030303031F1F1F1FF8F8F8F8C0C0C0C0C0C0C0C0F8F8F8
F8")
480 CALL CHAR(72,"0707070700000000000787877D7F7F3F1FFEFEFEFEF0F0F0F0F0F0F0F0F0E0
C")
490 CALL CHAR(76,"7878787879797F7F7F7F797978787878787878F0F0E0E0E0C0C0E0E0E0F0F078
78")
500 CALL CHAR(80,"787878787878787878787878787F7F7F7F000000000000000000000000F8F8F8
F8")
510 CALL CHAR(84,"F0F8FCFEFFFFF7F3F1F0F0F0F0F0F0F00F1F3F7FFFFFEFCF8F0F0F0F0F0F0F0F
0F")
520 CALL CHAR(88,"78787C7E7F7F7F7F7B7978787878787781E1E1E1E1E9EDEFEFEFEFE7E3E1E1E
1E")
530 CALL CHAR(92,"1F3F7F7F7C7878787878787C7F7F3F1FF8FCFEFE3E1E1E1E1E1E1E3EFEFEFC
F8")
540 CALL CHAR(96,"7F7F7F7F78787F7F7F7F787878787878F0FCFEFE1E1EFEFEFCF")
550 CALL CHAR(100,"1F3F7F7F7C78787B7B7B797C7F7F3F1FF8FCFEFE3E1E1E9EDEFEFEFEFEFCF
EEE")
560 CALL CHAR(104,"7F7F7F7F78787F7F7F7F787878787878F0FCFEFE1E1EFEFEFCF878783C3C1
E1E")
570 CALL CHAR(108,"1F3F7F7F78787F7F3F1F00007F7F7F7FFEFEFEFE0000F8FCFEFE1E1EFEFEF
CF8")
580 CALL CHAR(112,"7F7F7F7F0303030303030303030303FEFEFEFEC0C0C0C0C0C0C0C0C0C0C
0C")
590 CALL CHAR(116,"7878787878787878787878C7F7F3F0F1E1E1E1E1E1E1E1E1E1E1E3EFEFEF
CF")
600 CALL CHAR(120,"7878787878787C3C3E1E1F0F0F0703031E1E1E1E1E1E3E3C7C78F8F0F0E0C
0C")
```

```
610 CALL CHAR(124,"F0F0F0F0F0F0F0F1F3F7FFFFFEFCF8F00F0F0F0F0F0F0F8FCFEFFFFF7F3F1
F0F")
620 CALL CHAR(128,"787878787C3F3F1F1F3F3F7C787878781E1E1E1E3EFCFCF8F8FCFC3E1E1E1
E1E")
630 CALL CHAR(132,"78787878787C7F3F0F0303030303030031E1E1E1E1E3EFEFCF0C0C0C0C0C0C
0C")
640 CALL CHAR(136,"7F7F7F7F00000103070F1F3E7F7F7F7FFEFEFEFE7CF8F0E0C0800000FEFEF
EFE")
650 FOR I=1 TO 4 :: FOR J=1 TO 15 :: IF L(I,J)=0 THEN 670 :: X=((I-1)*3)+7 :: Y=
((J-1)*2)+2
660 CALL HCHAR(X,Y,L(I,J)):: CALL HCHAR(X+1,Y,L(I,J)+1):: CALL HCHAR(X,Y+1,L(I,J
)+2):: CALL HCHAR(X+1,Y+1,L(I,J)+3)
670 NEXT J :: NEXT I
680 CALL KEY(0,K,S):: IF S=0 THEN 680 :: CALL CLEAR :: CALL CHARSET
690 FOR I=1 TO 4 :: FOR J=1 TO 15 :: L(I,J)=0 :: NEXT J :: NEXT I :: GOTO 170
```

NUMEROLOGY

This program was written to demonstrate many of the features of TI Extended BASIC. By using subprograms, it was easier to write separate routines that would work by themselves. Then it was simple to put them together by adding extra subprograms such as delays, explanations and choices. This made the actual main program a mere six lines long (lines 140-190)!

If you are like most people, you may be too impatient to work with flow charts and start a program by writing it on paper or even typing it into the computer. It is easy to get bogged down in GOTO and GOSUB statements and lose track of where you are in a program. Subprograms can provide a more organized way of programming. For example, when you come to a stopping point, such as a delay, you can simply write CALL WAIT and go on with your programming. Then after you finish, you can add on the WAIT subprogram at the end. Another feature of TI Extended BASIC that adds convenience and speed is the DISPLAY AT statement, especially when you need to organize a screen full of data, such as in the CALL EXPLAIN subprogram used here.

The mathematical and string functions of the computer are ideal for determining the various numbers that a numerologist must calculate. Numerology is the study of occult significance of numbers as applied to the names and birthdates of individuals. Through this "Science of Numbers," one's strengths, weaknesses, emotions, talents, reactions and destiny can be determined. This program helps you to find the two most significant numbers used in numerology, the birthpath and destiny numbers. The birthpath is obtained by adding the eight digits of your birthdate. For example: December 23, 1964 (12/23/1964) would be 1+2+2+3+1+9+6+4=28. However, this number is not final because two digit numbers must be reduced until one digit is produced. Thus, in our example, 28 is broken into 2+8=10, then 10 becomes 1+0=1. The birthdate 12/23/1964 thus produces a birthpath of 1. There are three exceptions to this rule. The numbers 11, 22, and 33 are called "master numbers" and are not reduced to single digits. The

birthpath and destiny subprograms here include checks for master numbers to keep them from being reduced.

The destiny number is determined by assigning each of the letters of the alphabet a number of 1 to 9. The computer automatically performs this task in this program, then the numbers are added and the sum is reduced as in the birthpath routine. In order to simplify this program, the explanations for the birthpath and destiny numbers are the same. In reality, the interpretations may vary slightly. Also, there are other numbers that can be obtained by manipulating the letters of your name. For instance, there is the "Soul Number" that is derived by adding just the vowels in the name and the "Personality Number" that comes from adding the consonants. This program could be used to determine these numbers if you enter just those letters at the input prompt in the destiny routine. There are many other modifiers that can be derived from your name and birthdate, and since it is extremely rare for two people to have the exact same name and birthdate, the interpretations from numerology are very individualistic. This program presents a very basic introduction to numerology, but there are several books on the subject available for those desiring more information.

A point worth mentioning is that this program includes an example of how it is possible to perform a single function in more than one way in writing your programs. Line 500 of the birthpath routine uses a mathematical equation to reduce a two digit number to a one digit number. However, a similar reduction is performed in line 760 of the destiny routine by using a string function instead. Also, in writing this program the REDO function of Extended BASIC proved to be a time-saver in repeating many formats of the DISPLAY AT statements used in the EXPLAIN subprogram.

```
100 REM *NUMEROLOGY* TI EXTENDED BASIC
110 REM BY DON R. COOK
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 CALL TITLE
150 CALL CHOICE
160 CALL BIRTHPATH
170 CALL CHOICE
180 CALL DESTINY
190 CALL CHOICE
200 SUB TITLE
210 CALL CLEAR :: CALL SCREEN(10):: FOR A=1 TO 9 :: CALL MAGNIFY(2)
220 CALL SPRITE(#A,A+48,2,30,125,INT(RND*50-25),INT(RND*50-25)):: NEXT A
230 FOR B=2 TO 8 :: CALL COLOR(B,2,16):: NEXT B
240 DISPLAY AT(10,10):"NUMEROLOGY" :: DISPLAY AT(12,14):"BY" :: DISPLAY AT(14,10
):"DON R COOK" :: DISPLAY AT(18,8):"(PRESS ANY KEY)"
250 CALL DELAY :: CALL KEY(0,K,S):: IF S=0 THEN 250 :: CALL DELSPRITE(ALL)
260 FOR B=2 TO 8 :: CALL COLOR(B,2,1):: NEXT B
270 SUBEND
280 SUB CHOICE
290 CALL CLEAR :: CALL SCREEN(8)
300 DISPLAY AT(3,1):"WHICH NUMBER DO YOU WISH TO DETERMINE?" :: DISPLAY AT(7,1):
"1. BIRTHPATH NUMBER"
310 DISPLAY AT(10,1):"2. DESTINY NUMBER" :: DISPLAY AT(13,1):"3. END PROGRAM"
```

```
320 DISPLAY AT(22,1):"PRESS 1,2 OR 3 TO CONTINUE"
330 CALL DELAY :: CALL KEY(0,K,S):: IF S=0 OR K<49 OR K>51 THEN 330
340 IF K=49 THEN 160 :: IF K=50 THEN 180 :: IF K=51 THEN STOP
350 SUBEND
360 SUB BIRTHPATH
370 CALL CLEAR :: CALL SCREEN(4):: N,MONTH,DATE,YEAR=0
380 DISPLAY AT(3,5)ERASE ALL:"BIRTHDATE: " :: ACCEPT AT(3,16)VALIDATE(DIGIT)BEEP
 SIZE(8):A$
390 IF LEN(A$)<>8 THEN 400 ELSE 410
400 DISPLAY AT(12,1):"BIRTHDATE MUST BE IN 8-DIGITFORM":"FOR EXAMPLE: 01221950":
"FOR JANUARY 22,1950" :: CALL WAIT :: GOTO 380
410 MONTH=VAL(SEG$(A$,1,1))+VAL(SEG$(A$,2,1)):: DATE=VAL(SEG$(A$,3,1))+VAL(SEG$(
A$,4,1))
420 IF VAL(SEG$(A$,1,2))=11 THEN MONTH=11
430 IF VAL(SEG$(A$,3,2))=11 THEN DATE=11
440 IF VAL(SEG$(A$,3,2))=22 THEN DATE=22
450 FOR I=5 TO 8 :: YEAR=YEAR+VAL(SEG$(A$,I,1)):: NEXT I
460 N=MONTH+DATE+YEAR
470 IF N=0 THEN DISPLAY AT(10,1):"INVALID DATA--TRY AGAIN" :: CALL WAIT :: GOTO
380
480 IF N=11 OR N=22 OR N=33 THEN 510
490 IF N<=9 THEN 510
500 N=INT(N/10)+((N/10)-INT(N/10))*10 :: GOTO 480
510 DISPLAY AT(8,5):"BIRTHPATH: ";N :: DISPLAY AT(13,5):"DO YOU WANT AN EXPLANAT
ION  OF THIS NUMBER? (Y/N)"
520 CALL DELAY :: CALL KEY(0,K,S):: IF S=0 THEN 520
530 IF K=89 THEN CALL EXPLAIN(N)
540 DISPLAY AT(14,1)ERASE ALL:"DO YOU WISH TO DETERMINE":"ANOTHER BIRTHPATH? (Y/
N)"
550 CALL DELAY :: CALL KEY(0,K,S):: IF S=0 THEN 550
560 IF K=89 THEN 370 ELSE SUBEXIT
570 SUBEND
580 SUB DESTINY
590 CALL CLEAR :: CALL SCREEN(11):: N=0
600 DISPLAY AT(4,1)ERASE ALL:"FIRST NAME:" :: ACCEPT AT(4,15)VALIDATE(UALPHA)BEE
P:FN$
610 DISPLAY AT(6,1):"MIDDLE NAME:" :: ACCEPT AT(6,15)VALIDATE(UALPHA)BEEP:MN$
620 DISPLAY AT(8,1):"LAST NAME:" :: ACCEPT AT(8,15)VALIDATE(UALPHA)BEEP:LN$
630 A$=FN$&MN$&LN$ :: N=0
640 IF A$="" THEN DISPLAY AT(12,1):"INVALID DATA--TRY AGAIN" :: CALL WAIT :: GOT
O 600
650 DISPLAY AT(12,1):"ONE MOMENT--ANALYZING"
660 FOR I=1 TO LEN(A$):: L=65
670 X=1
680 FOR J=L TO L+8 :: IF L>90 THEN 720
690 IF SEG$(A$,I,1)=CHR$(J)THEN 710
700 X=X+1 :: NEXT J :: L=L+9 :: GOTO 670
710 N=N+X
720 NEXT I
730 DISPLAY AT(12,1):""
740 IF N=11 OR N=22 OR N=33 THEN 770
750 IF N>9 THEN 760 ELSE 770
760 N$=STR$(N):: N=VAL(SEG$(N$,1,1))+VAL(SEG$(N$,2,1)):: GOTO 740
770 DISPLAY AT(13,1):"DESTINY NUMBER:";N :: DISPLAY AT(18,5):"DO YOU WANT AN EXP
LANATION  OF THIS NUMBER? (Y/N)"
780 CALL DELAY :: CALL KEY(0,K,S):: IF S=0 THEN 780
790 IF K=89 THEN CALL EXPLAIN(N)
800 DISPLAY AT(14,1)ERASE ALL:"DO YOU WANT ANOTHER":"DESTINY NUMBER? (Y/N)"
810 CALL DELAY :: CALL KEY(0,K,S):: IF S=0 THEN 810
820 IF K=89 THEN 590
830 SUBEND
840 SUB WAIT
850 DISPLAY AT(24,1):"PRESS ANY KEY TO CONTINUE"
860 CALL DELAY :: CALL KEY(0,K,S):: IF S=0 THEN 860
870 SUBEND
```

```
880 SUB EXPLAIN(N)
890 IF N=11 THEN 910 :: IF N=22 THEN 960 :: IF N=33 THEN 1010
900 ON N GOTO 1060,1110,1160,1210,1260,1310,1360,1410,1460
910 DISPLAY AT(1,13)ERASE ALL:"11" :: DISPLAY AT(4,9):"ILLUMINATION" :: DISPLAY
AT(7,13):"(+)"
920 DISPLAY AT(9,1):"IDEALISTIC";TAB(15);"INTUITIVE" :: DISPLAY AT(10,1):"INSPIR
ATIONAL";TAB(15);"CAPABLE"
930 DISPLAY AT(11,1):"DREAMER";TAB(15);"VISIONARY" :: DISPLAY AT(12,1):"AVANT-GA
RDE";TAB(15);"CULTURED"
940 DISPLAY AT(14,13):"(-)" :: DISPLAY AT(16,1):"NERVOUS";TAB(15);"TENSE" :: DIS
PLAY AT(17,1):"IMPRACTICAL";TAB(15);"INTROVERTED"
950 DISPLAY AT(18,1):"SARCASTIC";TAB(15);"THOUGHTLESS" :: DISPLAY AT(19,1):"PESS
IMISTIC";TAB(15);"MATERIALISTIC" :: CALL WAIT :: SUBEXIT
960 DISPLAY AT(1,13)ERASE ALL:"22" :: DISPLAY AT(4,7):"MASTER BUILDER" :: DISPLA
Y AT(7,13):"(+)"
970 DISPLAY AT(9,1):"POWERFUL";TAB(15);"SUCCESSFUL" :: DISPLAY AT(10,1):"PRACTIC
AL";TAB(15);"PRESTIGIOUS"
980 DISPLAY AT(11,1):"CHARISMATIC";TAB(15);"UNORTHODOX" :: DISPLAY AT(12,1):"ART
ISTIC";TAB(15);"EMOTIONALLY-" :: DISPLAY AT(13,17):"CONTROLLED"
990 DISPLAY AT(15,13):"(-)" :: DISPLAY AT(17,1):"INSENSITIVE";TAB(15);"OVERBEARI
NG" :: DISPLAY AT(18,1):"DICTATORIAL";TAB(15);"OBSESSIONAL"
1000 DISPLAY AT(19,1):"HIGH-STRUNG";TAB(15);"MATERIALISTIC" :: CALL WAIT :: SUBE
XIT
1010 DISPLAY AT(1,13)ERASE ALL:"33" :: DISPLAY AT(4,9):"HUMANITARIAN" :: DISPLAY
 AT(7,13):"(+)"
1020 DISPLAY AT(7,1):"DEPENDABLE";TAB(15);"TRUSTING" :: DISPLAY AT(8,1):"LOVING"
;TAB(15);"COMPASSIONATE"
1030 DISPLAY AT(9,1):"EMOTIONAL";TAB(15);"HIGHLY-" :: DISPLAY AT(10,17):"SENSITI
VE" :: DISPLAY AT(11,1):"PROTECTIVE";TAB(15);"DEVOTED"
1040 DISPLAY AT(13,13):"(-)" :: DISPLAY AT(15,1):"INTROVERTED";TAB(15);"HIGH-STR
UNG" :: DISPLAY AT(16,1):"NERVOUS";TAB(15);"HIGHLY-"
1050 DISPLAY AT(17,17):"EMOTIONAL" :: DISPLAY AT(18,1):"DEPRESSED";TAB(15);"FRUS
TRATED" :: CALL WAIT :: SUBEXIT
1060 DISPLAY AT(1,14)ERASE ALL:"1" :: DISPLAY AT(4,7):"INDIVIDUALISTIC" :: DISPL
AY AT(7,13):"(+)"
1070 DISPLAY AT(9,1):"INDEPENDENT";TAB(15);"ORIGINAL" :: DISPLAY AT(10,1):"STRON
G";TAB(15);"COURAGEOUS"
1080 DISPLAY AT(11,1):"LEADER";TAB(15);"ACTIVE" :: DISPLAY AT(12,1):"AMBITIOUS";
TAB(15);"ACCOMPLISHED"
1090 DISPLAY AT(14,13):"(-)" :: DISPLAY AT(16,1):"STUBBORN";TAB(15);"SELFISH" ::
 DISPLAY AT(17,1):"BULLY";TAB(15);"DICTATORAL"
1100 DISPLAY AT(18,1):"EGOTISTICAL";TAB(15);"LAZY" :: CALL WAIT :: SUBEXIT
1110 DISPLAY AT(1,14)ERASE ALL:"2" :: DISPLAY AT(4,9):"COOPERATION" :: DISPLAY A
T(7,13):"(+)"
1120 DISPLAY AT(9,1):"PEACEFUL";TAB(15);"GENTLE" :: DISPLAY AT(10,1):"RESERVED";
TAB(15);"CONSIDERATE"
1130 DISPLAY AT(11,1):"MEDIATOR";TAB(15);"ANALYTICAL" :: DISPLAY AT(12,1):"SENSI
TIVE";TAB(15);"LOVING" :: DISPLAY AT(14,13):"(-)"
1140 DISPLAY AT(16,1):"INTROVERTED";TAB(15);"MOODY" :: DISPLAY AT(17,1):"PESSIMI
STIC";TAB(15);"OVERLY-" :: DISPLAY AT(18,17):"SENSITIVE"
1150 DISPLAY AT(19,1):"LETHARGIC";TAB(15);"DEPRESSED" :: CALL WAIT :: SUBEXIT
1160 DISPLAY AT(1,14)ERASE ALL:"3" :: DISPLAY AT(4,8):"JOY OF LIVING" :: DISPLAY
 AT(7,13):"(+)"
1170 DISPLAY AT(9,1):"HAPPY";TAB(15);"EFFERVESCENT" :: DISPLAY AT(10,1):"WARM";T
AB(15);"FRIENDLY"
1180 DISPLAY AT(11,1):"SOCIAL";TAB(15);"OPTIMISTIC" :: DISPLAY AT(12,1):"IMAGINA
TIVE";TAB(15);"CREATIVE" :: DISPLAY AT(14,13):"(-)"
1190 DISPLAY AT(16,1):"FRIVOLOUS";TAB(15);"SUPERFICIAL" :: DISPLAY AT(17,1):"VAI
N";TAB(15);"NOMADIC"
1200 DISPLAY AT(18,1):"ESCAPIST";TAB(15);"IRRESPONSIBLE" :: CALL WAIT :: SUBEXIT
1210 DISPLAY AT(1,14)ERASE ALL:"4" :: DISPLAY AT(4,11):"SERVICE" :: DISPLAY AT(7
,13):"(+)"
```

```
1220 DISPLAY AT(9,1):"PRACTICAL";TAB(15);"SYSTEMATIC" :: DISPLAY AT(10,1):"TRUST
WORTHTY";TAB(15);"DIGNIFIED"
1230 DISPLAY AT(11,1):"TENACIOUS";TAB(15);"SERIOUS" :: DISPLAY AT(12,1):"SINCERE
";TAB(15);"RESPONSIBLE" :: DISPLAY AT(14,13):"(-)"
1240 DISPLAY AT(16,1):"LIMITED";TAB(15);"FRUSTRATED" :: DISPLAY AT(17,1):"REPRES
SED";TAB(15);"OBSESSED"
1250 DISPLAY AT(18,1):"BOSSY";TAB(15);"NARROW-MINDED" :: CALL WAIT :: SUBEXIT
1260 DISPLAY AT(1,14)ERASE ALL:"5" :: DISPLAY AT(4,5):"CONSTRUCTIVE FREEDOM" ::
DISPLAY AT(7,13):"(+)"
1270 DISPLAY AT(9,1):"VERSATILE";TAB(15);"PROGRESSIVE" :: DISPLAY AT(10,1):"ENTH
USIASTIC";TAB(15);"ADVENTUROUS"
1280 DISPLAY AT(11,1):"CLEVER";TAB(15);"QUICK THINKER" :: DISPLAY AT(12,1):"TRAV
ELER";TAB(15);"FREEDOM-LOVER" :: DISPLAY AT(14,13):"(-)"
1290 DISPLAY AT(16,1):"INCONSISTANT";TAB(15);"TWO-FACED" :: DISPLAY AT(17,1):"SE
LFISH";TAB(15);"SELF-INDULGENT"
1300 DISPLAY AT(18,1):"AGGRESSIVE";TAB(15);"IRRESPONSIBLE" :: CALL WAIT :: SUBEX
IT
1310 DISPLAY AT(1,14)ERASE ALL:"6" :: DISPLAY AT(4,12):"HUMANE" :: DISPLAY AT(7,
13):"(+)"
1320 DISPLAY AT(9,1):"UNSELFISH";TAB(15);"LOVING" :: DISPLAY AT(10,1):"WISE & JU
ST";TAB(15);"RESPONSIBLE"
1330 DISPLAY AT(11,1):"SYMPATHETIC";TAB(15);"APPRECIATIVE" :: DISPLAY AT(12,1):"
TOLERANT";TAB(15);"PATIENT" :: DISPLAY AT(14,13):"(-)"
1340 DISPLAY AT(16,1):"NERVOUS";TAB(15);"CAUTIOUS" :: DISPLAY AT(17,1):"INTERFER
ING";TAB(15);"OVER-CRITICAL"
1350 DISPLAY AT(18,1):"JEALOUS";TAB(15);"PESSIMISTIC" :: CALL WAIT :: SUBEXIT
1360 DISPLAY AT(1,14)ERASE ALL:"7" :: DISPLAY AT(4,10):"ANALYTICAL" :: DISPLAY A
T(7,13):"(+)"
1370 DISPLAY AT(9,1):"INTROSPECTIVE";TAB(15);"RESERVED" :: DISPLAY AT(10,1):"DIG
NIFIED";TAB(15);"PEACEFUL"
1380 DISPLAY AT(11,1):"TRUSTWORTHY";TAB(15);"PERFECTIONIST" :: DISPLAY AT(12,1):
"INNER WISDOM";TAB(15);"AFFECTIONATE" :: DISPLAY AT(14,13):"(-)"
1390 DISPLAY AT(16,1):"LETHARGIC";TAB(15);"LACKADAISICAL" :: DISPLAY AT(17,1):"I
NCONSIDERATE";TAB(15);"QUARRELSOME"
1400 DISPLAY AT(18,1):"SARCASTIC";TAB(15);"SECRETIVE" :: CALL WAIT :: SUBEXIT
1410 DISPLAY AT(1,14)ERASE ALL:"8" :: DISPLAY AT(4,4):"MATERIAL SATISFACTION" ::
 DISPLAY AT(7,13):"(+)"
1420 DISPLAY AT(9,1):"POWERFUL";TAB(15);"SUCCESSFUL" :: DISPLAY AT(10,1):"EFFICI
ENT";TAB(15);"DEPENDABLE"
1430 DISPLAY AT(11,1):"REALISTIC";TAB(15);"PRACTICAL" :: DISPLAY AT(12,1):"FORCE
FUL";TAB(15);"CONFIDENT" :: DISPLAY AT(14,13):"(-)"
1440 DISPLAY AT(16,1):"MATERIALISTIC";TAB(15);"THOUGHTLESS" :: DISPLAY AT(17,1):
"INTERFERING";TAB(15);"INTOLERANT"
1450 DISPLAY AT(18,1):"LONELY";TAB(15);"OBSESSIVE" :: CALL WAIT :: SUBEXIT
1460 DISPLAY AT(1,14)ERASE ALL:"9" :: DISPLAY AT(4,9):"SELFLESSNESS" :: DISPLAY
AT(7,13):"(+)"
1470 DISPLAY AT(9,1):"GENEROUS";TAB(15);"COMPASSIONATE" :: DISPLAY AT(10,1):"IDE
ALISTIC";TAB(15);"PHILISOPHICAL"
1480 DISPLAY AT(11,1):"SENSITIVE";TAB(15);"ARTISTIC" :: DISPLAY AT(12,1):"HONORA
BLE";TAB(15);"SYMPATHETIC" :: DISPLAY AT(14,13):"(-)"
1490 DISPLAY AT(16,1):"INSINCERE";TAB(15);"INDISCREET" :: DISPLAY AT(17,1):"IMPU
LSIVE";TAB(15);"IMPATIENT"
1500 DISPLAY AT(18,1):"DICTATORIAL";TAB(15);"OVERLY-" :: DISPLAY AT(19,17):"EMOT
IONAL" :: CALL WAIT :: SUBEXIT
1510 SUBEND
1520 SUB DELAY
1530 FOR D=1 TO 30 :: NEXT D
1540 SUBEND
```

TAROT READER

For centuries, Tarot Card readings have been used by people seeking guidance in their lives. Usually, a deck of 78 special cards is shuffled while you concentrate on a question that you wish answered. A Tarot Card Reader turns over 10 cards, one at a time, and interprets the meaning of each card in regard to your question.

This program uses 74 cards (4 cards with obscure meanings were omitted). General definitions of the cards are used as this program is intended strictly for entertainment. The Extended BASIC Tarot Reader listing utilizes stationary sprites to cover and reveal the card graphics quickly. The computer draws a random design for the back of each card. To draw the faces of each card in sufficient detail would have required an enormous amount of memory and could not have run with just the console RAM. (If you use disk, you may have to type CALL FILES(1) before loading the program because of the length of the listing.)

Each time a card is selected, the meaning is read from Data statements. The computer checks to make sure that the card has not been previously used in the set of 10 cards currently being read. At the end of the reading, you are allowed to ask another question, or you may press Q (Quit) to end the program. Here's hoping your card reading says that good fortune is in store.

```
100 REM *TAROT READER* TI EXTENDED BASIC
110 REM BY BRIAN MADIGAN AND DAVID MIGICOVSKY
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 DIM A(10):: CALL CLEAR :: CALL SCREEN(2):: GOTO 870
150 DATA "THE FOOL - FOOLISH DECISIONSBUT POSSIBLY GREAT PROGRESS IF THE MATTER
IS SPIRITUAL."
160 DATA "THE MAGUS - THE POWER TO    ACHIEVE WHAT YOU DESIRE IS  IN YOUR HANDS,
BUT--USE IT   WISELY."
170 DATA "THE HIGH PRIESTESS - THE    ABILITY TO USE YOUR POWERS  OF ATTRACTION
TO ACHIEVE    YOUR  DESIRES."
180 DATA "THE EMPRESS - MOTHERLY,     WISE,INSPIRING DEDICATION   AND LOVE IN TH
OSE AROUND HER"
190 DATA "THE EMPEROR - THE GREAT KINGLEADING HIS PEOPLE, POWER,  MIGHT, ENERGY.
 A LEADER IN  BATTLE AND PEACE."
200 DATA "THE HIEROPHANT - COMPLETION OF THE TASK AT HAND, PROPER AND EXACT USE
OF POWER. ABI-LITY TO SOLVE YOUR PROBLEMS THROUGH THOUGHT."
210 DATA "THE LOVERS - LOVE BOTH      MUNDANE AND SPIRITUAL.      UNION, MARRIAG
E AND TOTAL   GIVING OF ONESELF."
220 DATA "THE CHARIOT - WAR, ENERGY,  FIGHTING; TIME OF VICTORIOUSBATTLE. FIGHTI
NG TO DESTROY.SUDDEN AND DRASTIC CHANGES. TRAVELLING"
230 DATA "ADJUSTMENT - A TIME OF      BALANCING OUT IS AT HAND.   AN ATTEMPT TO
SEE THINGS    MORE CLEARLY AND STRAIGHTEN OUT PRIORITIES."
240 DATA "THE HERMIT - SEARCHING, NOT FOR SOMETHING LOST BUT FOR  SOMETHING NEVE
R HAD.        A QUEST."
250 DATA "FORTUNE - THE EQUILIBRIUM OFCYCLES. EVENTS WILL FOLLOW  IN A NATURAL A
ND PREDICTABLECOURSE."
260 DATA "STRENGTH - POWER. GOALS CAN BE ATTAINED BY THE USE OF   BRUTE FORCE."
270 DATA "THE HANGED MAN - LIFE IS IN SUSPENSION A PERIOD OF WAIT-ING AND INERTI
A WHICH MAY BEQUITE TRYING, OUTSIDE INFLU-ENCES AT HAND."
```

```
280 DATA "DEATH - A GREAT CHANGE OF   DRASTIC PROPORTION--          NOT NECESSARIL
Y PHYSICAL    DEATH. A SHOCKING CHANGE    IN OUTLOOK."
290 DATA "ART-THE ABILITY TO TAKE THE NECESSARY STEPS TO CREATE,  USING A PROPER
 MIXTURE OF   QUALITIES TO ACHIEVE THE    DESIRED END."
300 DATA "THE DEVIL - OVER-ATTACHMENT TO PHYSICAL MATTERS. BEING  HELD BACK FROM
 PROGRESS BY  PHYSICAL DESIRES. SLAVERY   IN GENERAL."
310 DATA "THE TOWER - SUDDEN VIOLENT  DESTRUCTION BRINGING ABOUT  A GREAT CHANGE
 AND PEACE.   THE VIOLENT END OF A BAD    SITUATION."
320 DATA "THE STAR - JOY, HAPPINESS,  FULFILLMENT OF YOUR DESTINY."
330 DATA "THE MOON - THERE IS A       DECEPTION THAT YOU MIGHT NOTRECOGNIZE UNTI
L TOO LATE.   MAY ALSO MEAN A NARROWLY    ESCAPED DECEPTION."
340 DATA "THE SUN - JOY, HEALTH, AND  SUCCESS. SPIRITUAL          ADVANCEMENT. T
HE ESSENCE OF PERSONAL INDIVIDUALITY."
350 DATA "THE AEON - FLOWING WITH THE NATURAL COURSE OF THE       UNIVERSE. ALL
DESIRES CAN BEFULFILLED. YOU ARE ON THE   RIGHT TRACK."
360 DATA "THE UNIVERSE - ALL          POSSIBILITIES ARE STILL OPENTO YOU. MANY D
ECISIONS MUST BE MADE."
370 DATA "THE KNIGHT OF WANDS - AN    OLDER MAN IN YOUR LIFE.     GENEROUS, FIER
CE, PROUD AND UNPREDICTABLE."
380 DATA "THE QUEEN OF WANDS - AN     OLDER  WOMAN IN YOUR LIFE.  PERSISTENT AND
 CALM. KINDLY,BUT IMPATIENT. MAY BECOME   TYRANNICAL."
390 DATA "THE PRINCE OF WANDS -  A    YOUNG MAN IN YOUR LIFE.     SWIFT AND  STR
ONG, BUT CAN  BE IMPULSIVE. OPINIONATED   AND VIOLENT."
400 DATA "THE PRINCESS OF WANDS - A   YOUNG WOMAN IN  YOUR LIFE.  EXTREMELY INDI
VIDUAL.       BRILLIANT AND DARING.       EMOTIONAL."
410 DATA "OPPRESSION - BEING HELD     BACK. OUTER INFLUENCES ARE  UN-COOPERATIVE
. OPPOSITION  IN GENERAL."
420 DATA "POWER - OUTGOING AND FORCE- FUL. DEDICATION, ACHIEVEMENTAND VICTORY."
430 DATA "SWIFTNESS - THINGS ARE      MOVING VERY QUICKLY, PERHAPSMORE SWIFTLY T
HAN YOU ARE   ABLE TO COPE WITH. SUDDEN   CHANGE."
440 DATA "VALOR - BEING TRUE TO ONE-  SELF. YOU ARE NOT WILLING TOCOMPROMISE YOU
R BELIEFS TO  ACHIEVE YOUR GOAL."
450 DATA "VICTORY - THE ABILITY TO    OVERCOME ALL OPPOSITION.    BATTLES CAN BE
 WON EASILY."
460 DATA "STRIFE - POVERTY,           LONELINESS. OFTEN CAUSED BY OVER-USE OF A
SHARP TONGUE."
470 DATA "COMPLETION - PRESENT CYCLES ARE COMING TO AN END,       OPENING THE WA
Y TO A NEW    BEGINNING."
480 DATA "VIRTUE - A TIME OF HARMONY, PEACE AND NEW BEGINNINGS."
490 DATA "DOMINION - RULERSHIP,       CONTROL OF YOUR LIFE AND OF OTHER PEOPLE."
500 DATA "KNIGHT OF CUPS - AN OLDER   MAN IN YOUR LIFE. GRACEFUL, AMIABLE AND PA
SSIVE. SENSUALAND SENSITIVE, MAY BE IDLE  AND UNTRUTHFUL."
510 DATA "QUEEN OF CUPS - ON OLDER    WOMAN IN YOUR LIFE. DREAMY, TRANQUIL MOTHE
RLY AND       EMOTIONAL. EASILY INFLUENCEDBY OTHERS."
520 DATA "PRINCE OF CUPS - A YOUNG MANIN YOUR LIFE. SUBTLE,       SECRETIVE AND
CRAFTY. A     PASSIONATE ARTIST WHO CAN BEPOWER HUNGRY"
530 DATA "PRINCESS OF CUPS - A YOUNG  WOMAN IN YOUR LIFE. SWEET,  GRACIOUS AND R
OMANTIC, BUT  MAY APPEAR SELFISH."
540 DATA "SATIETY - FULFILLMENT. YOU  HAVE OBTAINED ALL YOU CAN   FROM THE PRESE
NT SITUATION."
550 DATA "HAPPINESS - CONTENTMENT IN  YOUR LIFE AND YOUR PRESENT  SITUATION."
560 DATA "INDOLENCE - YOU DO NOT WANT TO MAKE REQUIRED EFFORT TO  ACHIEVE YOUR G
OAL. LAZINESS."
570 DATA "DEBAUCHERY - TOO MUCH OF A  GOOD THING. YOU HAVE BITTEN OFF MORE THAN
YOU CAN CHEW."
580 DATA "PLEASURE - EMOTIONAL AND    PHYSICAL ENJOYMENT."
590 DATA "DISAPPOINTMENT - UNPLEASANT SURPRISES AND AN UNFORTUNATEOUTCOME TO THE
 PRESENT      SITUATION."
600 DATA "LUXURY - YOU WILL ACHIEVE   EVEN MORE THAN YOU DESIRE INYOUR PRESENT S
ITUATION."
610 DATA "LOVE - A RENEWED SPARK IN   YOUR PRESENT RELATIONSHIP,  OR THE START O
F  A NEW MORE FULFILLING ONE."
620 DATA "KNIGHT OF SWORDS - AN OLDER MAN IN YOUR LIFE. SKILLED,  SUBTLE AND CLE
VER. HE IS    COURAGEOUS, DELICATE AND    FIERCE."
```

```
630 DATA "QUEEN OF SWORDS - AN OLDER  WOMAN IN YOUR LIFE. SHE IS  INTENSELY PERC
EPTIVE, A GOODOBSERVER AND INTERPRETER."
640 DATA "PRINCE OF SWORDS  - HE IS   ABSTRACT THINKER WHO REDUCESEVERYTHING TO
UNREALITY. HE IS NOT ALWAYS TRUSTWORTHY."
650 DATA "RUIN - TOTAL DESTRUCTION ANDLOSS IN ANY SITUATION. CAN  ALSO MEAN RUIN
 OF AN ENEMY."
660 DATA "CRUELTY - UNJUST PUNISHMENT AND UNFAIRNESS."
670 DATA "INTERFERENCE - OUTSIDE      INFLUENCES ARE INTERRUPTING YOUR EFFORTS A
ND CLOUDING   YOUR GOALS."
680 DATA "FUTILITY - DESIRES CANNOT BEFULFILLED AT THE MOMENT.    STOP PUSHING."
690 DATA "SCIENCE - ANALYTICAL,       LOGICAL AND ORGANIZED       THOUGHT IS REQ
UIRED TO REACHTHE DESIRED END."
700 DATA "DEFEAT - OPPOSITION IS TOO  STRONG. YOU DO NOT HAVE THE RESOURCES TO F
IGHT IT."
710 DATA "TRUCE - A SHORT-LIVED CALM  AMIDST CONFUSION. A TIME FORRE-ORGANIZATIO
N AND THOUGHT."
720 DATA "SORROW - PAIN AND EMOTIONAL LOSS."
730 DATA "PEACE - THE CALM AFTER THE  STORM."
740 DATA "KNIGHT OF DISCS - AN OLDER  MAN IN YOUR LIFE.LARGE. VERYOCCUPIED WITH
MATERIAL      THINGS. LABORIOUS AND       PATIENT."
750 DATA "QUEEN OF DISCS  - AMBITIOUS,AFFECTIONATE AND KIND. HAS  GREAT INTUITIO
N; MOTHERLY."
760 DATA "PRINCE OF DISCS - ENERGETIC,ENDURING, TRUSTWORTHY AND   ADAPTABLE. LAC
KING IN EMO-  TION AND SENSITIVITY."
770 DATA "PRINCESS OF DISCS - A YOUNG WOMAN DREAMING OF HEARTH ANDHOME."
780 DATA "WEALTH - UNEXPECTED INCOME."
790 DATA "GAIN -  A PERMANENT         IMPROVEMENT IN YOUR FIN-    ANCIAL  STATUS
."
800 DATA "PRUDENCE - CAREFUL ANALYSIS OF ALL POSSIBILITIES IS     REQUIRED, BUT
DON'T OVERDO  IT."
810 DATA "FAILURE"
820 DATA "SUCCESS - YOU WILL MEET WITHSUCCESS IN ALL YOUR PHYSICALENDEAVORS."
830 DATA "WORRY - BE CAREFULL. YOU    MUST OVERCOME YOUR WORRY BE-FORE YOU CAN S
EE CLEARLY."
840 DATA "POWER - THE ABILITY TO BUY  YOUR WAY IN TO  ANY         SITUATION."
850 DATA "ENDEAVORS - PUTTING THE     VARIOUS ASPECTS  OF YOUR    PLANS SUCCESSF
ULLY INTO     EFFECT."
860 DATA "CHANGE - SLOW AND POSITIVE  ADVANCEMENT."
870 RANDOMIZE :: FOR S1=0 TO 8
880 III=INT(RND*14)+3 :: CALL COLOR(S1,III,1,9,7,1,10,7,1):: NEXT S1
890 DISPLAY AT(10,1):"WHAT IS YOUR NAME? " :: ACCEPT AT(12,1)SIZE(17)BEEP:N$
900 DISPLAY AT(14,1):"HELLO ";N$;", ";"I AM MADAM ";"ZOLTANA, THE GREAT SEER." :
: GOTO 920
910 CALL DELSPRITE(ALL):: CALL CLEAR
920 DISPLAY AT(18,1):"WHAT IS YOUR QUESTION?"
930 DISPLAY AT(13,13):"" :: DISPLAY AT(24,1):"AT ANY TIME, PRESS Q TO QUIT"
940 ACCEPT AT(20,1)SIZE(16)BEEP:Q$ :: CALL CLEAR :: IF Q$="Q" THEN END
950 DISPLAY AT(1,12):N$
960 IF Q$="" THEN DISPLAY AT(3,12):"" ELSE DISPLAY AT(3,12):Q$;"?"
970 GOSUB 1060
980 FOR I=1 TO 10
990 L=INT(RND*68)+1
1000 FOR II=1 TO 10 :: IF A(II)=L THEN 990
1010 NEXT II
1020 A(I)=L
1030 GOSUB 1190
1040 NEXT I
1050 GOTO 910
1060 P=0 :: CALL CHAR(96,"0000000000000000000FF80402018040101010202020204040 4FF
0808081010")
```

```
1070 CALL CHAR(100,"0201000000000000000001010101000010A06030584442818284182040
8")
1080 CALL CHAR(104,"8080404040404040202020FF10101008080000000000000000000FF010204
1820")
1090 CALL CHAR(108,"0805060C12224281412118040201000040800000000000000000080808080
")
1100 CALL HCHAR(6,20,96):: CALL HCHAR(7,20,97):: CALL HCHAR(6,21,98):: CALL HCHA
R(7,21,99):: CALL HCHAR(8,20,100):: CALL HCHAR(9,20,101)
1110 CALL HCHAR(8,21,102):: CALL HCHAR(9,21,103):: CALL HCHAR(6,22,104):: CALL H
CHAR(7,22,105):: CALL HCHAR(6,23,106):: CALL HCHAR(7,23,107)
1120 CALL HCHAR(8,22,108):: CALL HCHAR(9,22,109):: CALL HCHAR(8,23,110):: CALL H
CHAR(9,23,111)
1130 CALL CHAR(120,"C1C1C1C1C1C1C1C1838383838383838383")
1140 CALL CHAR(128,"FFFFE0C0C0C0C0C0FFFF0000000000FFFF0000000000FFFFC0C0C0C0C0E0
FFFF")
1150 CALL HCHAR(1,3,128):: CALL HCHAR(1,4,129,8):: CALL CHAR(132,"FFFF0703030303
0303030303030307FFFF"):: CALL HCHAR(1,12,132):: CALL VCHAR(2,12,121,13)
1160 CALL HCHAR(15,3,131):: CALL HCHAR(15,4,130,8):: CALL HCHAR(15,12,133):: CAL
L VCHAR(2,3,120,13)
1170 CALL CHAR(112,"0"):: FOR Q=1 TO 8 :: P=P+1 :: CALL VCHAR(2,P+3,112,13):: NE
XT Q
1180 RETURN
1190 FOR K=1 TO L
1200 READ DESC$
1210 NEXT K
1220 IF I=1 THEN I$="POS. INFLUENCES" ELSE IF I=2 THEN I$="NEG. INFLUENCES" ELSE
 IF I=3 THEN I$="       PAST"
1230 IF I=4 THEN I$="     PRESENT" ELSE IF I=5 THEN I$="     OUTLOOK" ELSE IF I=
6 THEN I$="     FUTURE"
1240 IF I=7 THEN I$="       YOU" ELSE IF I=8 THEN I$="     OTHERS" ELSE IF I=9 T
HEN I$="INNER FEELINGS" ELSE IF I=10 THEN I$="     OUTCOME"
1250 I2=INT(RND*10)+1
1260 IF I2=1 THEN CALL CHAR(112,"3536584643373454")ELSE IF I2=2 THEN CALL CHAR(1
12,"ACDFB5DE2354ADFC")
1270 IF I2=3 THEN CALL CHAR(112,"245764CA53BC83875")ELSE IF I2=4 THEN CALL CHAR(
112,"7656FAED654346548")
1280 IF I2=5 THEN CALL CHAR(112,"4365DFECA7543152")ELSE IF I2=6 THEN CALL CHAR(1
12,"107E42DA5B427E08")
1290 CALL SOUND(-366,(RND*200)+110,0,(RND*300)+110,8,(RND*800)+110,25)
1300 IF I2=7 THEN CALL CHAR(112,"9942249918244299")ELSE IF I2=8 THEN CALL CHAR(1
12,"CCCC3333CCCC3333")
1310 IF I2=9 THEN CALL CHAR(112,"183C7EFFFF7E3C18")ELSE IF I2=10 THEN CALL CHAR(
112,"1824428181422418")
1320 CALL SOUND(-866,(RND*4345)+110,0,(RND*100)+110,25,(RND*33345)+110,10)
1330 III=INT(RND*14)+3 :: II=INT(RND*14)+3
1340 DISPLAY AT(18,1):"" :: DISPLAY AT(19,1):"" :: DISPLAY AT(20,1):"" :: DISPLA
Y AT(21,1):"" :: DISPLAY AT(22,1):""
1350 CALL COLOR(11,II,III,12,2,16,13,2,16):: CALL DELSPRITE(#9,#8,#7):: CALL DEL
SPRITE(#6,#5,#4,#3,#2,#1)
1360 CALL SOUND(-400,(RND*9500)+110,0,(RND*6000)+110,0,(RND*1000)+110,8)
1370 DISPLAY AT(13,12):I$ :: DISPLAY AT(16,4):I :: DISPLAY AT(18,1):DESC$
1380 CALL SOUND(-540,(RND*505)+110,0,(RND*2000)+110,5,(RND*100)+110,25)
1390 DISPLAY AT(24,2):"PRESS ANY KEY TO CONTINUE" :: CALL KEY(0,K,S):: Z=RND ::
IF S=0 THEN 1390 :: IF K=81 THEN 1460
1400 CALL COLOR(13,1,1,12,1,1,11,1,1):: CALL MAGNIFY(4):: CALL SPRITE(#1,140,2,1
,15)
1410 CALL SPRITE(#2,140,2,1,44,#3,140,2,1,73,#4,140,2,33,12,#5,140,2,33,44,#6,14
0,2,33,76,#7,140,2,65,12)
1420 CALL SPRITE(#8,140,2,65,44,#9,140,2,65,76)
1430 CALL CHAR(140,"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFF")
1440 RESTORE
1450 RETURN
1460 CALL CLEAR :: END
```

TALKING TI TAROT

Here is a program that actually makes your computer a Tarot Card READer. Madame Zoltana, our local seer, resides in RAM, ready to speak the sooth, the whole sooth and nothing but the sooth! This TI BASIC version of the Tarot Card reading program has been written especially for use with the Terminal Emulator II module and the TI Speech Synthesizer. Madame Zoltana receives her information not from another world but rather from Data statements which have been revised to make pronunciation easier for her. (She gets very nervous if people make fun of her slight speech impediment). Because of the length of some of the data, you may need to use the Insert and Edit functions of the computer to type the long Data statements. (TI BASIC normally accepts only statements up to 4 screen lines long, but sometimes you can edit a line and insert characters to make it longer. Or, you could key in the listing in Extended BASIC, which accepts longer program lines.) If you use disk, you will need to type CALL FILES(1) to release enough memory before loading the program because of the length of the program. Be sure to have the Terminal Emulator II and Speech Synthesizer connected when you run the program so that you won't get reprimanding file error messages from the mysterious forces within the computer.

```
100 REM * TALKING TI TAROT * TI BASIC
110 REM BY DAVID MIGICOVSKY AND BRIAN MADIGAN
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 REM TERMINAL EMULATOR 2 AND SPEECH SYNTHESIZER REQUIRED
150 RANDOMIZE
160 CALL SCREEN(12)
170 CALL CLEAR
180 OPEN #1:"SPEECH",OUTPUT
190 PRINT #1:"//50 160"
200 PRINT "      TALKING TI TAROT"
210 PRINT
220 PRINT "  FEATURING MADAME ZOLTANA"
230 FOR X=1 TO 10
240 PRINT
250 NEXT X
260 PRINT #1:"TALKING T I TARROE"
270 PRINT #1:"FEATCHURING MADDAM ZOLTANNUH"
280 DIM A(10)
290 DATA "THE FOOL. FOOL ISH  DECISIONS BUT  POSSIBLY GREAT  PROGRESS  IF THE M
ATTER IS SPIRITUAL!"
300 DATA "THE  MAG US. THE  POWER TO  ACHIEVE  WHAT YOU DIZZIRE  IS IN YOR HANDS
; BUT USE IT WIZE LEE!"
310 DATA "THE HI PREESS TESS. TH E UH BILLITY TO USE YOR POWERS OF ATTRACTION TO
 ACHIEVE YOR DIZZIRES."
320 DATA "TH E EM PRUSS. MUH THERLY, WISE, INSPIRING DEDICATION AND LUV IN THOSE
 AROUND HER"
```

```
330 DATA "TH E EM PER RER. A GREAT KEENG LEADING HIS PEEPL. POWER; MITE; AND ENE
RGY"
340 DATA "THE HI RO FUNT. COMPLEETION OF THE TASK AT HAND, PROPPER AND EXACT USE
 OF POWER. UH BILLITY TO SOLVE YOUR PROBLEMS THREW THTHOTT."
350 DATA "THE LOVERS. LUV BOTH MUNDANE AND SPIRITUAL. UNION )MARRIAGE AND TOTAL
  GIVING OF WON SELF."
360 DATA "THE CHARRY UT. WAR; ENERGY; FITING; TIME OF VICTOARR EE USS BATTL. FIT
ING TO DESTROY. SUDDEN AND DRASTIC CHANGES. )TRAVV LLING"
370 DATA "ADJUSTMENT. A TIME OF BALENCING OUT IS AT HAND. AN ATTEMPT TO SEE THIN
GS MOR CLEARLY AND STRATE EN OUT PRY ORITIES."
380 DATA "THE HERMIT. SEARCHING; NOT FOR SUMTHING LOST; BUT FOR SUMTHING NEVER H
AD! UH QUEST."
390 DATA "FORTCH UN. THE EQUAL IBBREE UHM OF CYKE LS. EVENTS WILL FOLLOE IN UH N
ATURAL AND PREDICT UHBL COURSE."
400 DATA "STRAYNGTH. POWER. GOLES CAN BE ATTAINED; BY THE USE OF BROOT FORCE."
410 DATA "THE. HANGED MAN. LIFE IS IN SUSPENSION; A PERIOD OF WAITING AND INNERS
HA WHICH MAY BE QUITE TRYING; OUTSIDE INFLUENCES AT HAND."
420 DATA "DEATH. UH GREAT CHANGE OF DRASSTIC PROPORTION. NOT NAE CESSARILY FIZZI
CAL DEATH. A SHOCKING CHANGE IN OUTLOOK."
430 DATA "ART. TH E UH BILLITY TO TAKE THE NAE CESSARY STEPS TO CREATE; USING A
PROPPER MIXTURE OF QUAULITTEES TO ACHEEVE THE DIZZIRED END."
440 DATA "THE DEVVLL. OVER ATTACHMENT TO FIZZICAL MATTERS; BE ING HELD BACK FROM
 PROG RESS BY FIZZICAL DESIRES. SLAY VRE IN GENERAL."
450 DATA "THE TOWER. SUDDEN )VIALENT DESTRUCTION BRING ING ABOUT UH GREAT CHANGE
 AND PEACE. THE )VIALENT END OF A BAD SITUATION."
460 DATA "THE STAR. JOY AND HAPPEENESS; FULL FILLMENT OF YOR DESTINY."
470 DATA "THE MOON. THER IS UH DECEPTION THAT YOU MITE NOT REC OGNIZE UNTIL TO L
ATE. MAY ALSO MEAN A NARROE LEE ESCAPED DECEPTION."
480 DATA "THE SUN. JOY; HELTH, AND SUCCESS. SPIRITUAL ADVANCEMENT. THE ESSENCE O
F PERSUNNAL INDIVI DUALITY."
490 DATA "TH E A YON. FLOE ING WITH THE NATURAL COURSE OF THE UNIVERSE. ALL DESI
RES CAN BE FULL FILLED. YOU ARE ON THE RITE TRACK."
500 DATA "THE UNIVERSE. ALL POSSUH BILLITIES ARE STILL OPENTO YOU. MANY DECISION
S MUST BE MADE."
510 DATA "THE NITE OF WAUNDS. AN OLDER MAN IN YOUR LIFE.JENNER US; FIERCE. PROUD
 AND UNPREDICTABLE."
520 DATA "THE QUEEN OF WAUNDS. AN OLDER  WUHMUN IN YOUR LIFE.  PERSISTENT AND CO
M. KINDLY; BUT IMM PAY SHENT. MAY BECOME TIR ANNICAL."
530 DATA "THE PRINCE OF WAUNDS. UH YUNG MAN IN YOUR LIFE. SWIFT AND STRONG, BUT
CAN  BE IMM PUL SIV. OPIN YUN ATED AND )VIALENT."
540 DATA "THE PRINCESS OF WAUNDS. A YUNG WUHMUN IN YUR LIFE. EX TREME LEE INDIVI
DUAL. BRILLIANT AND DARING. EMOTIONAL."
550 DATA "OPPRETION. BEING HELD BACK. OUTER INFLUENCES ARE UN CO OPPERATIVE. OPP
OSITION IN GENERAL."
560 DATA "POWER. OUT GOING AND FORCE FUL. DEDDICATION, ACHIEVEMENT AND VICTORY."
570 DATA "SWIFTNESS. THINGS ARE MOVING VERY QUICKLY, PERHAPS MOR SWIFTLY THAN YO
U ARE ABLE TO COPE WITH. SUDDEN CHANGE."
580 DATA "VALOR. BEING TRUE TO WUN SELF. U ARE NOT WILLING TO COMPROMISE YOR BEL
EEFS TO  ACHEEVE YOR GOLE."
590 DATA "VICTORY. TH E UH BILLITY TO OVER CUM ALL OPPOSITION. BATT LS CAN BE WO
N EASILY."
600 DATA "STRIFE. POVERTY, LONE LEENESS. OFFEN COZZED BY OVER USE OF UH SHARP TU
NG."
610 DATA "COMPLEETION. PRESENT CYKE LS ARE CUMMING TO AN END, OPEN ING THE WAY T
O A NEW BEGINNING."
620 DATA "VIRCHU. UH TIME OF HARMUNNY; PEACE AND NEW BE GINNINGS."
630 DATA "DOMINION. RULERSHIP, CONTROL OF YOR LIFE AND OF UTHER PEEPLE."
640 DATA "NITE OF CUPS. AN OLDER MAN IN YOR LIFE. GRACE FUL, AMIABLE AND PASSIV.
 SENSHUAL AND SENSITIV, MAY BE I DL AND UNTRUTHFUL."
650 DATA "QUEEN OF CUPS. AN OLDER WUHMUN IN YOR LIFE. DREAMY, TRANQUIL MUH THERL
Y AND EMOTIONAL. EASILY INFLUENCED BY OTHERS."
660 DATA "PRINCE OF CUPS. UH YOUNG MAN IN YOR LIFE. SUTTL, SEE CRETTIV AND CRAFT
Y. UH PASHUNNIT ARTIST WHO CAN BE POWER HUNGRY"
670 DATA "PRINCESS OF CUPS. UH YUNG WUHMUN IN YOR LIFE. SWEET GRAY SHUSS AND RO
MANTIC; BUT MAY APPEAR SELFISH."
```

```
680 DATA "SAY SHUTTY. FULL FILLMENT. YOU HAV OBTAINED ALL YOU CAN FROM THE PREZZ
ENT SITUATION."
690 DATA "HAPPINESS. CONTENTMENT IN  YOR LIFE AND YOR PREZZENT SITUATION."
700 DATA "UH BUNDUNCE. A GOOD CONCLUSION TO ANY MATTER. VICTORY; SOL US; HEALING
"
710 DATA "IN DULLENCE. YOU DO NOT WANT TO MAKE THE RE QUIRED EFFORT TO  ACHEEVE
YOUR GOLE. LAZEE NESS."
720 DATA "DE BOTCHURRY. TO MUCH OF A GOOD THING. YOU HAVE BIT EN OFF MOR THAN U
CAN CHEW."
730 DATA "PLEASURE. EMOTIONAL AND FIZZICAL ENJOYMENT."
740 DATA "DISSAPPOINTMENT. UNPLEZUNT SURPRISES AND AN UNFORTCHUNIT OUTCOME TO TH
E PREZZENT SITUATION."
750 DATA "LUG ZHURY. YOU WILL ACHIEVE EVEN MORE THAN YOU DESIRE IN YOUR PRESENT
SITUATION."
760 DATA "LOVE. UH RENEWD SPARK IN YOUR PRESENT RELATIONSHIP; OR THE START OF UH
 NEW MORE FULL FILLING ONE."
770 DATA "KNIGHT OF SORDS. AN OLDER MAN IN YOUR LIFE. SKILLED; SUTTL AND CLEVVER
. HE IS CURR A JUSS, DELICUT AND FIERCE."
780 DATA "QUEEN OF SORDS. AN OLDER WUHMUN IN YOUR LIFE; SHE IS INTENSSLY PERCEPT
IVE; A GOOD OBSERVER AND INTER PRET ER."
790 DATA "PRINCE OF SORDS. HE IS AN ABSTRACT THEENKER WHO REDUCES EVERYTHING TO
UN RE ALLITY. HE IS NOT OLLWAYS TRUSTWORTHY."
800 DATA "PRINCESS OF SORDS. A YUNG WUHMUN YOU NO. AA LERT; LYE TH AND INTELLECT
UAL. MAY POSESS UH CUTTING WIT"
810 DATA "RUIN. TOTAL DESTRUCTION AND LOSS IN ANY SITUATION; CAN ALSO MEAN RUIN
OF AN ENEMY."
820 DATA "CROOLTY. UNJUST PUNNISHMENT AND UNFAIRNESS."
830 DATA "INTERFEARENCE. OUTSIDE INFLUENCES ARE INTER UPTING YOUR EFFORTS AND CL
OUDING YOUR GOLES."
840 DATA "FUE TILLITY. DESIRES CANNOT BE FULL FILLED AT THE MOMENT; STOP PUSHING
."
850 DATA "SIE YENCE. ANNUL ITTICAL, LOJUHCL AND ORGAN IZED THOT IS REQUIRED TO R
EACHTHE DESIRED END"
860 DATA "DE FEAT. OPPOSITION IS TOO STRONG. YOU DO NOT HAVE THE RESOURCES TO FI
GHT IT."
870 DATA "TRUCE. UH SHORT LIVED COM AMMIDST CONFUSION; A TIME FOR RE ORGANIZATIO
N AND THOT."
880 DATA "SORROE. PAIN AND EMOTIONAL LOSS."
890 DATA "PEACE. THE COM AFTER THE STORM."
900 DATA "NIGHT OF DISCS. AN OLDER MAN IN YOUR LIFE; LARGE. VERY OCCUPIED WITH M
ATTEARIAL THINGS. LAA BORY YUS AND PAY SHUNT."
910 DATA "QUEEN OF DISCS. AMBISHUS, AFFECTIONATE AND KIND. HAS GREAT INTUITION;
MUHTHERLY."
920 DATA "PRINCE OF DISCS. ENNER JETTIC; ENDURING; TRUSTWORTHY AND ADAPTABLE. LA
CKING IN EMOTION AND SENSITIVITY."
930 DATA "PRINCESS OF DISCS. UH YUNG WUHMUN DREAMING OF HARTH AND HOME."
940 DATA "WELTH. UNEXPECTED INCOME."
950 DATA "GAIN. UH PERMANENTIMPROOV MENT IN YOUR FINE ANCIAL STATTIS."
960 DATA "PRUE DENCE. CAREFUL UH NAL ISSIS OF ALL POSSIBILITIES IS REQUIRED; BUT
 DOANT OVER DO IT."
970 DATA "FAIL YER"
980 DATA "SUCCESS. YOU WILL MEET WITH SUCCESS IN ALL YOUR FIZZICAL ENDEVVORS."
990 DATA "WORRY. BE CAREFULL. YOU MUST OVER COME YOUR WORRY BE FOR YOU CAN SEE C
LEARLY."
1000 DATA "POWER. TH E UHBILLITY TO BY YOUR WAY IN TO ANY SITUATION."
1010 DATA "ENDEVVORS. PUT ING THE VARY US ASPECTS  OF YOUR PLANS SUCCESSFULLY IN
 TO EFFECT."
1020 DATA "CHANGE. SLOW AND POSITIVE ADVANCEMENT."
1030 PRINT #1:"//25 80"
1040 FOR S1=1 TO 10
1050 CALL COLOR(S1,7,1)
1060 NEXT S1
1070 PRINT #1:"WHAT IS YOUR NAME?"
1080 INPUT "":N$
1090 CALL CLEAR
```

```
1100 PRINT #1:"HELLO; ";N$;"; I AM MADDAM ";"ZOLTANNUH, THE GREAT SEAR."
1110 CALL CLEAR
1120 CALL SCREEN(12)
1130 PRINT #1:"WHAT IS YOUR QUESTION? "
1140 INPUT "":Q$
1150 Q$=Q$&"?"
1160 CALL CLEAR
1170 CALL SCREEN(2)
1180 PRINT #1:Q$
1190 PRINT TAB(14-LEN(N$)/2);N$: :TAB(14-LEN(Q$)/2);Q$: : :"AT ANY TIME PRESS Q
TO QUIT"
1200 GOSUB 1310
1210 FOR I=1 TO 10
1220 L=INT(RND*74)+1
1230 FOR II=1 TO 10
1240 IF A(II)=L THEN 1220
1250 NEXT II
1260 A(I)=L
1270 GOSUB 1850
1280 NEXT I
1290 GOTO 1110
1300 P=0
1310 CALL CHAR(97,"0000FF8040201804")
1320 CALL CHAR(98,"0101020202020204")
1330 CALL CHAR(99,"0404FF0808081010")
1340 CALL CHAR(100,"0201")
1350 CALL CHAR(101,"000001010101")
1360 CALL CHAR(102,"10A0603058444281")
1370 CALL CHAR(103,"82841820408")
1380 CALL CHAR(104,"808040404040402")
1390 CALL CHAR(105,"2020FF1010100808")
1400 CALL CHAR(107,"0000FF0102041820")
1410 CALL CHAR(108,"0805060C12224281")
1420 CALL CHAR(109,"412118040201")
1430 CALL CHAR(110,"408")
1440 CALL CHAR(111,"000080808080")
1450 CALL HCHAR(7,20,97)
1460 CALL HCHAR(6,21,98)
1470 CALL HCHAR(7,21,99)
1480 CALL HCHAR(8,20,100)
1490 CALL HCHAR(9,20,101)
1500 CALL HCHAR(8,21,102)
1510 CALL HCHAR(9,21,103)
1520 CALL HCHAR(6,22,104)
1530 CALL HCHAR(7,22,105)
1540 CALL HCHAR(7,23,107)
1550 CALL HCHAR(8,22,108)
1560 CALL HCHAR(9,22,109)
1570 CALL HCHAR(8,23,110)
1580 CALL HCHAR(9,23,111)
1590 FOR X=11 TO 13
1600 CALL COLOR(X,1,1)
1610 NEXT X
1620 CALL CHAR(120,"C1C1C1C1C1C1C1C1")
1630 CALL CHAR(121,"83838383838383838383")
1640 CALL CHAR(128,"FFFFE0C0C0C0C0C0")
1650 CALL CHAR(129,"FFFF0000000000FF")
1660 CALL CHAR(130,"FF000000000FFFF")
1670 CALL CHAR(131,"C0C0C0C0C0E0FFFF")
1680 CALL CHAR(132,"FFFF07030303030303")
1690 CALL CHAR(133,"030303030307FFFF")
1700 CALL HCHAR(1,3,128)
1710 CALL HCHAR(1,4,129,8)
1720 CALL HCHAR(1,12,132)
1730 CALL VCHAR(2,12,121,13)
```

```
1740 CALL HCHAR(15,3,131)
1750 CALL HCHAR(15,4,130,8)
1760 CALL HCHAR(15,12,133)
1770 CALL VCHAR(2,3,120,13)
1780 CALL CHAR(112,"0")
1790 P=0
1800 FOR Q=1 TO 8
1810 P=P+1
1820 CALL VCHAR(2,P+3,112,13)
1830 NEXT Q
1840 RETURN
1850 FOR K=1 TO L
1860 READ DESC$
1870 ON I GOTO 1880,1900,1920,1940,1960,1980,2000,2020,2040,2060
1880 I$="CARD 1; POSUHTIV INFLUENCES"
1890 GOTO 2070
1900 I$="CARD 2; NEGUHTIV INFLUENCES"
1910 GOTO 2070
1920 I$="CARD 3; PAST"
1930 GOTO 2070
1940 I$="CARD 4; PRESENT"
1950 GOTO 2070
1960 I$="CARD 5; OUTLOOK"
1970 GOTO 2070
1980 I$="CARD 6; FU CHUR"
1990 GOTO 2070
2000 I$="CARD 7; U"
2010 GOTO 2070
2020 I$="CARD 8; OTHERS"
2030 GOTO 2070
2040 I$="CARD 9; INNER FEELINGS"
2050 GOTO 2070
2060 I$="CARD TEN; OUTCUM"
2070 NEXT K
2080 XX=INT(RND*10)+1
2090 ON XX GOTO 2100,2120,2140,2160,2180,2200,2220,2240,2260,2280
2100 CALL CHAR(112,"3536584643373454")
2110 GOTO 2290
2120 CALL CHAR(112,"ACDFB5DE2354ADFC")
2130 GOTO 2290
2140 CALL CHAR(112,"245764CA53BC83875")
2150 GOTO 2290
2160 CALL CHAR(112,"7656FAED654346548")
2170 GOTO 2290
2180 CALL CHAR(112,"4365DFECA7543152")
2190 GOTO 2290
2200 CALL CHAR(112,"107E42DA5B427E08")
2210 GOTO 2290
2220 CALL CHAR(112,"9942249918244299")
2230 GOTO 2290
2240 CALL CHAR(112,"CCCC3333CCCC3333")
2250 GOTO 2290
2260 CALL CHAR(112,"183C7EFFFF7E3C18")
2270 GOTO 2290
2280 CALL CHAR(112,"1824428181422418")
2290 III=INT(RND*15)+2
2300 II=INT(RND*15)+2
2310 IF III=II THEN 2290
2320 CALL COLOR(11,II,III)
2330 CALL COLOR(12,2,16)
2340 CALL COLOR(13,2,16)
2350 PRINT #1:I$
2360 PRINT #1:DESC$
2370 PRINT #1:"//50 160"
2380 IF I>9 THEN 2430
```

```
2390 FOR D=1 TO 50
2400 NEXT D
2410 PRINT #1:"PRESS UH KEY PLEAZ"
2420 GOTO 2440
2430 PRINT #1:"PRESS ANY KEY TO ASK UN OTHER QUESTION; OR PRESS Q TO QUIT"
2440 CALL KEY(0,K,S)
2450 X=RND
2460 IF S=0 THEN 2440
2470 PRINT #1:"//25 80"
2480 IF K=81 THEN 2540
2490 CALL COLOR(13,1,1)
2500 CALL COLOR(12,1,1)
2510 CALL COLOR(11,1,1)
2520 RESTORE
2530 RETURN
2540 CALL CLEAR
2550 PRINT "GOOD-BYE FROM MADAME ZOLTANA"
2560 PRINT #1:"GOOD BYE FROM MADDAM ZOLTANNUH"
2570 FOR X=1 TO 24
2580 PRINT
2590 NEXT X
2600 CLOSE #1
2610 END
```

SPRITE DANCE

The movement of "dancers" on the screen makes this music program more entertaining than a program that just plays a tune while you stare at a blank screen, and it provides a good example of how music and sprites can be used together in TI Extended BASIC. The music was given priority, so animation had to be altered if it interfered with the flow of the music. Though the characters may not move in absolutely perfect rhythm, their movement should be close enough to give the desired effect. Repeated CALL SOUND statements were used, rather than reading the note values from Data statements, because it allows the music to play more smoothly and it was easier to time the sprite pattern changes.

Notice that the graphics routines take up relatively little program space, demonstrating how the TI Extended BASIC sprite commands make simple work out of tasks that would be either difficult, impossible or require machine code on some other computers. If you do not yet have TI Extended BASIC, you might want to type in just the Call Sound statements as single statement lines and eliminate the graphics subroutines. This would allow you to at least hear the music. Then, perhaps you could design your own TI BASIC graphics to watch while listening.

```
100 REM *SPRITE DANCE* TI EXTENDED BASIC
110 REM BY MIKE WILCOX
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 CALL CLEAR :: CALL SCREEN(16):: CALL CHAR(139,"F0F0F0F00F0F0F0F")
150 FOR I=1 TO 32 :: CALL HCHAR(1,I,139):: NEXT I
160 FOR I=1 TO 24 :: CALL VCHAR(I,32,139):: NEXT I
170 FOR I=32 TO 1 STEP -1 :: CALL HCHAR(24,I,139):: NEXT I
180 FOR I=24 TO 1 STEP -1 :: CALL VCHAR(I,1,139):: NEXT I
190 DISPLAY AT(8,7):"* SPRITE DANCE *" :: DISPLAY AT(10,8):"BY MIKE WILCOX"
```

```
200 DISPLAY AT(18,4):"PRESS ANY KEY TO BEGIN" :: DISPLAY AT(22,7):"(COPYRIGHT 19
83)"
210 DEF XX=INT(14*RND+2)
220 CALL KEY(0,K,S):: CALL COLOR(14,XX,1):: CALL COLOR(14,1,XX):: IF S=0 THEN 22
0
230 CALL CLEAR :: CALL MAGNIFY(3):: CALL SCREEN(16)
240 CALL CHAR(128,"000101000103050909010101010380C0C080C0E0D0C8C8C0C04040404
060")
250 CALL CHAR(132,"000101000103050911010102040830008C0C080C0E0D0C8C4C0C04040404
060")
260 CALL CHAR(136,"000101201F0101010101010101010380C0C084F8C0C0C0C0C0C04040404
060")
270 CALL CHAR(140,"040909040301010101010101010101390C8C890E0C0C0C0C0C0C02010080
600")
280 CALL CHAR(96,"020505040301010101010101010101A0D0D090E0C0C0C0C0C0C040201008
00")
290 CALL CHAR(104,"020505040301010101010101020408000A0D0D090E0C0C0C0C0C0C04040404
040")
300 FOR A=1 TO 8 :: CALL SPRITE(#A,128,1+A,A*20,124,0,(A*2)):: NEXT A
310 RP=RP+1
320 CALL SOUND(235,262,1):: CALL SOUND(235,330,2):: CALL SOUND(235,392,2):: CALL
 SOUND(235,330,2)
330 GOSUB 1640
340 CALL SOUND(235,523,0):: CALL SOUND(235,392,2):: CALL SOUND(235,330,2):: CALL
 SOUND(235,392,2)
350 GOSUB 1650
360 IF RP<3 THEN 310
370 CALL SOUND(235,262,2):: CALL SOUND(235,330,2):: CALL SOUND(235,392,2):: CALL
 SOUND(235,330,2):: CALL SOUND(235,523,1)
380 GOSUB 1660
390 CALL SOUND(235,392,2):: CALL SOUND(235,330,2):: CALL SOUND(235,392,2,523,0):
: CALL SOUND(235,262,2,523,0)
400 GOSUB 1670
410 CALL SOUND(235,330,2,392,0):: CALL SOUND(235,392,2,523,0):: CALL SOUND(235,5
23,1,659,0):: CALL SOUND(235,330,2,523,0)
420 GOSUB 1680
430 CALL SOUND(235,392,2,659,0):: CALL SOUND(235,523,1,784,0):: CALL SOUND(235,3
92,2,523,0):: CALL SOUND(235,349,2,1047,0)
440 GOSUB 1690
450 CALL SOUND(235,440,2,988,0):: CALL SOUND(235,523,2,880,0):: CALL SOUND(235,6
98,2,784,0):: CALL SOUND(235,262,2,784,1)
460 GOSUB 1630
470 CALL SOUND(235,330,2,784,2):: CALL SOUND(235,392,2):: CALL SOUND(235,523,1)
480 GOSUB 1640
490 CALL SOUND(235,196,2,784,0):: CALL SOUND(235,247,2,698,0):: CALL SOUND(235,2
94,2,587,0):: CALL SOUND(235,392,2,494,0)
500 GOSUB 1650
510 CALL SOUND(235,247,2,392,0):: CALL SOUND(235,294,2,494,0):: CALL SOUND(235,3
92,2,587,0):: CALL SOUND(235,294,2,784,0)
520 GOSUB 1660
530 CALL SOUND(235,262,2,659,0):: CALL SOUND(235,330,2,523,0):: CALL SOUND(235,5
23,1,880,0):: CALL SOUND(235,392,2,784,0)
540 GOSUB 1670
550 CALL SOUND(235,330,2,784,1):: CALL SOUND(235,392,2,784,2):: CALL SOUND(235,5
23,1):: CALL SOUND(235,392,2,523,0)
560 GOSUB 1680
570 CALL SOUND(235,262,2,523,0):: CALL SOUND(235,330,2,392,0):: CALL SOUND(235,3
92,2,523,0):: CALL SOUND(235,523,1,659,0)
580 GOSUB 1690
590 CALL SOUND(235,330,2,523,0):: CALL SOUND(235,392,2,659,0):: CALL SOUND(235,5
23,2,784,0):: CALL SOUND(235,392,2,523,0)
600 GOSUB 1630
610 CALL SOUND(235,349,2,1047,0):: CALL SOUND(235,440,2,988,0):: CALL SOUND(235,
523,2,880,0):: CALL SOUND(235,698,2,784,0)
620 GOSUB 1640
```

```
630 CALL SOUND(235,262,2,784,1):: CALL SOUND(235,330,2,784,2):: CALL SOUND(235,3
92,2):: CALL SOUND(235,523,2)
640 GOSUB 1650
650 CALL SOUND(235,196,2,784,0):: CALL SOUND(235,247,2,698,0):: CALL SOUND(235,2
94,2,587,0):: CALL SOUND(235,392,2,494,0)
660 GOSUB 1660
670 CALL SOUND(235,247,2,392,0):: CALL SOUND(235,294,2,494,0):: CALL SOUND(235,3
92,2,587,0):: CALL SOUND(235,294,2,494,0)
680 GOSUB 1670
690 CALL SOUND(235,262,2,523,0):: CALL SOUND(235,330,2,392,0):: CALL SOUND(235,3
92,2,659,0):: CALL SOUND(235,523,2,523,0)
700 GOSUB 1680
710 CALL SOUND(235,330,2,523,1):: CALL SOUND(235,392,2,523,2):: CALL SOUND(235,5
23,2):: CALL SOUND(235,196,2,523,0)
720 GOSUB 1690
730 CALL SOUND(235,131,2,523,0):: CALL SOUND(235,131,2,196,2,392,0):: CALL SOUND
(235,262,2,330,2,523,0)
740 GOSUB 1630
750 CALL SOUND(235,262,2,330,2,659,0):: CALL SOUND(235,131,2,196,2,523,0):: CALL
 SOUND(235,131,2,196,2,659,0)
760 GOSUB 1640
770 CALL SOUND(235,262,2,330,2,784,0):: CALL SOUND(235,131,2,523,0)
780 GOSUB 1650
790 CALL SOUND(235,1047,0):: CALL SOUND(235,175,2,988,0):: CALL SOUND(235,220,2,
262,2,880,0):: CALL SOUND(235,196,2,784,0)
800 GOSUB 1660
810 CALL SOUND(235,131,2,784,1):: CALL SOUND(235,196,2,784,2):: CALL SOUND(235,2
62,2,330,2):: CALL SOUND(235,147,2)
820 GOSUB 1670
830 CALL SOUND(235,1568,0):: CALL SOUND(235,196,2,1397,0):: CALL SOUND(235,247,2
,349,2,1175,0)
840 GOSUB 1680
850 CALL SOUND(235,247,2,349,2,988,0):: CALL SOUND(235,196,2,784,0):: CALL SOUND
(235,196,2,988,0)
860 GOSUB 1690
870 CALL SOUND(235,247,2,349,2,1175,0):: CALL SOUND(235,196,2,1397,0):: CALL SOU
ND(235,131,2,1319,0)
880 GOSUB 1630
890 CALL SOUND(235,131,2,196,2,1047,0):: CALL SOUND(235,262,2,330,2,1760,0):: CA
LL SOUND(235,262,2,330,2,1568,0)
900 GOSUB 1640
910 CALL SOUND(235,196,2,15688,1):: CALL SOUND(235,196,2,1568,2):: CALL SOUND(23
5,262,2,330,2):: CALL SOUND(235,196,2,523,0)
920 GOSUB 1650
930 CALL SOUND(235,131,2,523,0):: CALL SOUND(235,131,2,196,2,392,0):: CALL SOUND
(235,262,2,330,2,523,0)
940 GOSUB 1660
950 CALL SOUND(235,262,2,330,2,659,0):: CALL SOUND(235,196,2,523,0):: CALL SOUND
(235,196,2,659,0)
960 GOSUB 1670
970 CALL SOUND(235,262,2,330,2,784,0):: CALL SOUND(235,131,2,523,0):: CALL SOUND
(235,1047,0):: CALL SOUND(235,175,2,988,0)
980 GOSUB 1680
990 CALL SOUND(235,220,2,262,2,880,0):: CALL SOUND(235,196,2,784,0):: CALL SOUND
(235,131,2,784,1)
1000 GOSUB 1690
1010 CALL SOUND(235,196,2,784,2):: CALL SOUND(235,262,2,330,2):: CALL SOUND(235,
147,2)
1020 GOSUB 1630
1030 CALL SOUND(235,1568,0):: CALL SOUND(235,196,2,1397,0):: CALL SOUND(235,247,
2,349,2,1175,0):: CALL SOUND(235,247,2,349,2,988,0)
1040 GOSUB 1640
1050 CALL SOUND(235,196,2,784,0):: CALL SOUND(235,196,2,988,0):: CALL SOUND(235,
247,2,349,2,1175,0):: CALL SOUND(235,196,2,988,0)
1060 GOSUB 1650
```

```
1070 CALL SOUND(235,131,2,1047,0):: CALL SOUND(235,131,2,196,2,784,0):: CALL SOU
ND(235,262,2,330,2,1319,0)
1080 GOSUB 1660
1090 CALL SOUND(235,262,2,330,2,1047,0):: CALL SOUND(235,196,2,1047,1):: CALL SO
UND(235,196,2,1047,0)
1100 GOSUB 1670
1110 CALL SOUND(235,262,2,330,2,1047,0):: CALL SOUND(235,196,2,1047,0):: CALL SO
UND(235,1397,2,1760,1,2094,0)
1120 GOSUB 1680
1130 CALL SOUND(235,175,2,1760,0):: CALL SOUND(235,220,2,262,2,1397,0):: CALL SO
UND(235,220,2,262,2,1047,0)
1140 GOSUB 1690
1150 CALL SOUND(235,175,2,880,0):: CALL SOUND(235,175,2,1047,0):: CALL SOUND(235
,220,2,262,2,1397,0)
1160 GOSUB 1630
1170 CALL SOUND(235,196,2,1760,0):: CALL SOUND(235,131,2,1568,0):: CALL SOUND(23
5,131,2,196,2,1047,0)
1180 GOSUB 1640
1190 CALL SOUND(235,262,2,330,2,1760,0):: CALL SOUND(235,262,2,330,2,1568,0):: C
ALL SOUND(235,196,2,1568,1)
1200 GOSUB 1650
1210 CALL SOUND(235,196,2,1568,2):: CALL SOUND(235,262,2,330,2):: CALL SOUND(235
,147,2,196,2)
1220 GOSUB 1660
1230 CALL SOUND(235,988,2,1175,1,1568,0):: CALL SOUND(235,196,2,1397,0):: CALL S
OUND(235,294,2,349,2,1175,0)
1240 GOSUB 1670
1250 CALL SOUND(235,294,2,349,2,988,0):: CALL SOUND(235,196,2,784,0):: CALL SOUN
D(235,247,2,988,0)
1260 GOSUB 1680
1270 CALL SOUND(235,249,2,294,2,1175,0):: CALL SOUND(235,131,2,1319,0):: CALL SO
UND(235,131,2,262,2,1047,0)
1280 GOSUB 1690
1290 CALL SOUND(235,659,2,523,2,1760,0):: CALL SOUND(235,659,2,523,2,1568,0)
1300 GOSUB 1630
1310 CALL SOUND(235,262,2,1568,1):: CALL SOUND(235,262,2,1319,0):: CALL SOUND(23
5,659,2,523,2,1319,0)
1320 GOSUB 1640
1330 CALL SOUND(235,131,2,175,2,1319,0):: CALL SOUND(235,1397,2,1760,1,2049,0)::
 CALL SOUND(235,175,2,1760,0)
1340 GOSUB 1650
1350 CALL SOUND(235,220,2,262,2,1397,0):: CALL SOUND(235,220,2,262,2,1047,0):: C
ALL SOUND(235,175,2,880,0)
1360 GOSUB 1660
1370 CALL SOUND(235,175,2,1047,0):: CALL SOUND(235,220,2,262,2,1397,0):: CALL SO
UND(235,196,2,1760,0)
1380 GOSUB 1670
1390 CALL SOUND(235,131,2,1568,0):: CALL SOUND(235,131,2,196,2,1047,0):: CALL SO
UND(235,262,2,330,2,1760,0)
1400 GOSUB 1690
1410 CALL SOUND(235,262,2,330,2,1568,0):: CALL SOUND(235,196,2,1047,0):: CALL SO
UND(235,262,2,330,2,1047,0)
1420 GOSUB 1630
1430 CALL SOUND(235,196,2,1047,0)
1440 GOSUB 1640
1450 CALL SOUND(235,988,2,1175,1,1568,0):: CALL SOUND(235,196,2,1397,0):: CALL S
OUND(235,247,2,349,2,1175,0)
1460 GOSUB 1650
1470 CALL SOUND(235,247,2,349,2,988,0):: CALL SOUND(235,196,2,784,0):: CALL SOUN
D(235,196,2,988,0)
1480 GOSUB 1660
1490 CALL SOUND(235,247,2,349,2,1175,0):: CALL SOUND(235,196,2,988,0):: CALL SOU
ND(235,131,2,1047,0)
1500 GOSUB 1670
```

```
1510 CALL SOUND(235,131,2,196,2,784,0):: CALL SOUND(235,262,2,330,2,1319,0):: CA
LL SOUND(235,196,2,1047,0)
1520 GOSUB 1680
1530 CALL SOUND(500,262,2,330,2,1047,0):: CALL SOUND(500,40000,30)
1540 GOSUB 1690
1550 TIME=TIME+1 :: IF TIME<2 THEN RP=0 :: GOTO 310
1560 CALL CLEAR :: FOR D=1 TO 100 :: NEXT D
1570 CALL DELSPRITE(ALL)
1580 DISPLAY AT(5,1):"IF YOU WOULD LIKE TO HEAR IT" :: DISPLAY AT(7,1):"AGAIN PR
ESS ""Y"",IF NOT PRESS"
1590 DISPLAY AT(9,1):"ANY OTHER KEY."
1600 CALL KEY(3,K,S):: IF S=0 THEN 1600
1610 IF K=89 THEN TIME,RP=0 :: CALL CLEAR :: GOTO 300
1620 CALL CLEAR :: END
1630 CALL PATTERN(#1,128,#2,128,#3,128,#4,128,#5,128,#6,128,#7,128,#8,128):: RET
URN
1640 CALL PATTERN(#1,132,#2,132,#3,132,#4,132,#5,132,#6,132,#7,132,#8,132):: RET
URN
1650 CALL PATTERN(#1,136,#2,136,#3,136,#4,136,#5,136,#6,136,#7,136,#8,136):: RET
URN
1660 CALL PATTERN(#1,140,#2,140,#3,140,#4,140,#5,140,#6,140,#7,140,#8,140):: RET
URN
1670 CALL PATTERN(#1,96,#2,96,#3,96,#4,96,#5,96,#6,96,#7,96,#8,96):: RETURN
1680 CALL PATTERN(#1,104,#2,104,#3,104,#4,104,#5,104,#6,104,#7,140,#8,104):: RET
URN
1690 CALL PATTERN(#1,136,#2,136,#3,136,#4,136,#5,136,#6,136,#7,136,#8,136):: RET
URN
```

ADVENTURE IN OZ

Inspired by the Oz books of L. Frank Baum, this game differs from most adventure-type games in that, like the "good" characters in the Oz stories, you do not get killed. Rather, you take a bizarre journey through a strange land. Your object is to find a way to return to your home in Kansas in as few turns as possible, but a number of random events may slow your progress and cause you to lose turns. Too much should not be said about the game because, like the Land of Oz itself, it contains a few surprises. However, a brief explanation of the structure of the program may help you understand how it works.

Because of the length of the program, Memory Expansion and Disk are required. To utilize Memory Expansion as well as sprite graphics, the program is written is TI Extended BASIC. The TI 99/4 stores your program listing and values of numeric variables in the 24K portion of the Memory Expansion. Values of string variables are stored in VDP (console RAM). Long strings were needed to keep track of events that would occur at the 768 different locations in the Land of Oz as well as to define some graphics characters. As the program began to exceed the limits of Memory Expansion, it was necessary to create data files that could be read off of disk and stored in VDP RAM. It was also necessary to separate much of the music into another program which could be chained to the main program with the RUN statement.

Listed here are actually 4 programs, *Adventure In Oz,* which is the main program of the game, *OZFILE1* and *OZFILE2,* which create the 2 data files that are read by the main program, plus *Rainbow,* which is a separate music program that is run by the main program. *Rainbow* and the data files (OZDATA1 and OZDATA2) should be saved on the same disk as the game, *Adventure In Oz.* Thus, if you also save the programs that create the files, your disk catalog would look like this:

FILENAME	SIZE	TYPE
OZ	88	INT/VAR254
OZDATA1	5	INT/FIX 40
OZDATA2	8	INT/FIX 20
OZFILE1	8	PROGRAM
OZFILE2	11	PROGRAM
RAINBOW	16	PROGRAM

To run the main program (if saved as "OZ"), then you would just type RUN "DSK1.OZ" and leave the disk in the drive so that the program can access the other files automatically.

Because there are 768 screen locations (24 rows by 32 columns), the map of Oz used here (which is based on a color map issued by the International Wizard of Oz Club) uses all these locations as places that you can move in your adventure. Any time you try to go into a forbidden area, such as the impassable desert, which surrounds the land, the program puts you back to the previous space and increments your number of turns used. To determine your location at any time, you may press M (for Map) and a graphic display of your whereabouts will be presented. However, each time you do this, you will be using up a turn.

To keep track of the many different locations and events in the game, the program is made up almost entirely of subroutines and subprograms. In fact, the main program section is only 5 lines long, beginning at line 230. Each time you move, the main routine checks your current row and column position. The value of the variable X is derived by selecting the string of characters for your row (R$(R)), and counting up to the particular character in that string that corresponds to your column, taking the ASCII code of that letter, and subtracting 64. This yields a number between 1 and 26 which can be used in an ON GOSUB statement to branch to the appropriate routine for each location. Additionally, a random number between 0 and 9 is generated each turn so that an event from several possible random events may be selected within each routine.

Placing Data statements for each routine within the routines themselves made the program self-documenting and easier to write. Only one DISPLAY AT statement was used to display the hundreds of lines of text contained in the Data statements. Each event restores a particular Data statement then branches to a routine that

reads the starting row and column and the number of lines of text to be displayed. The text is made invisible by use of a CALL COLOR statement, then the text is displayed in a FOR-NEXT loop. Next, another CALL COLOR statement makes the text appear all at once on the screen. Using this method of display allowed a great amount of versatility in formatting the text on the screen.

Color plays a very important role in the magic countries in the Land of Oz. Each country has its favorite color, denoted on the map in the program. To keep track of screen colors, which let you know which country you are in, each event occurs only in the North and South countries or in the East and West countries. Two subroutines automatically set the proper screen and text foreground colors based upon your row and column position. Routines that are accessed only a few times in the program, such as the title screen and instructions, were put into subprograms so that they could be placed at the end of the listing to speed program execution.

So, if you are ready to get away from it all, pack your broomstick, take a trip to everyone's favorite magic kingdom, and enjoy your Adventure in Oz. But just remember, there's no place like home!

```
100 REM * ADVENTURE IN OZ * TI EXTENDED BASIC
110 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
120 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
130 REM REQUIRES MEMORY EXPANSION & DISK FILES OZDATA1,OZDATA2 & RAINBOW
140 OPTION BASE 1 :: DIM R$(24),W$(54),WD$(19):: RANDOMIZE
150 OPEN #1:"DSK1.OZDATA1",INPUT ,INTERNAL,FIXED 40 :: FOR I=1 TO 24 :: INPUT #1
:R$(I):: NEXT I
160 CLOSE #1
170 OPEN #2:"DSK1.OZDATA2",INPUT ,INTERNAL,FIXED 20 :: FOR I=1 TO 19 :: INPUT #2
:WD$(I):: NEXT I
180 FOR I=1 TO 54 :: INPUT #2:W$(I):: NEXT I
190 CLOSE #2
200 CALL TITLE :: CALL CYCLONE :: CALL RAINBOW :: CALL DELSPRITE(ALL):: CALL INS
TRUCT
210 R=INT(RND*22)+2 :: C=INT(RND*30)+2 :: X=ASC(SEG$(R$(R),C,1))-64 :: IF X=10 O
R X=11 OR X=15 OR X=18 OR X=19 THEN 230 ELSE 210
220 LR=R-1 :: LC=C-1
230 X=ASC(SEG$(R$(R),C,1))-64 :: Y=INT(RND*10):: IF X>13 THEN 260
240 ON X GOSUB 480,530,570,610,620,670,750,790,900,980,1050,1150,1190
250 GOTO 270
260 ON X-13 GOSUB 1210,1260,1310,1370,1550,1600,1700,1750,1790,1810,1880,1990,20
40
270 TURN=TURN+1 :: GOTO 230
280 DISPLAY AT(22,2):"IT TOOK YOU ";STR$(TURN);" TURNS!" :: CALL SAY("GOODBYE PL
EASE COME+BACK+AGAIN SOME+TIME")
290 RUN "DSK1.RAINBOW"
300 IF C>19 THEN SCR=5 ELSE IF C<12 THEN SCR=12 ELSE SCR=13
310 IF SCR<>12 THEN F=16 ELSE F=2
320 CALL CLEAR :: CALL SCREEN(SCR):: RETURN
330 IF R<11 THEN SCR=14 ELSE IF R>14 THEN SCR=7 ELSE SCR=13
340 F=16 :: CALL CLEAR :: CALL SCREEN(SCR):: RETURN
350 LR=R :: LC=C
360 CALL KEY(0,K,S):: IF S=0 THEN 360 ELSE IF POS("NSEWMQ",CHR$(K),1)=0 THEN CAL
L SOUND(50,110,2):: GOTO 360
370 IF K=78 AND R=1 OR K=83 AND R=24 OR K=69 AND C=32 OR K=87 AND C=1 THEN 380 E
LSE 400
```

```
380 DISPLAY AT(1,1):"  YOU CANNOT GO THAT WAY!    " :: CALL SOUND(150,110,0):: FO
R D=1 TO 500 :: NEXT D
390 DISPLAY AT(1,1):"" :: TURN=TURN+1 :: GOTO 360
400 IF K=78 THEN R=R-1 ELSE IF K=83 THEN R=R+1
410 IF K=87 THEN C=C-1 ELSE IF K=69 THEN C=C+1
420 IF K=77 THEN CALL MAP(R,C)
430 IF K=81 THEN 2250
440 RETURN
450 CALL COLOR(1,1,1,2,1,1,3,1,1,4,1,1,5,1,1,6,1,1,7,1,1,8,1,1):: READ PR,PC,N
460 FOR I=1 TO N :: READ X$ :: DISPLAY AT(PR,PC):X$ :: PR=PR+2 :: NEXT I
470 CALL COLOR(1,F,1,2,F,1,3,F,1,4,F,1,5,F,1,6,F,1,7,F,1,8,F,1):: RETURN
480 GOSUB 300 :: CALL CHAR(96,"04081828C7"):: IF C<12 THEN CALL COLOR(9,15,6)ELS
E CALL COLOR(9,5,8)
490 CALL HCHAR(14,1,96,352):: IF Y>6 THEN 500 ELSE RESTORE 510 :: GOSUB 450 :: R
=LR :: C=LC :: CALL WAIT :: RETURN
500 RESTORE 520 :: GOSUB 450 :: GOSUB 350 :: RETURN
510 DATA 3,2,3,THE WIDE WINDING WINKIE,RIVER IS TOO WIDE TO CROSS,BETTER GO ANOT
HER WAY
520 DATA 3,2,4,AT THE BANK OF THE BLUE,WATER A LITTLE MAN WITH A,FERRY BOAT OFFE
RS TO HELP,YOU CROSS THE WIDE RIVER
530 GOSUB 330 :: RESTORE 550 :: GOSUB 450 :: FOR I=200 TO 800 STEP 50 :: CALL SO
UND(-150,I,0):: NEXT I :: CALL SOUND(500,110,0)
540 GOSUB 350 :: TURN=TURN+1 :: RETURN
550 DATA 2,1,9,THE CHINA COUNTRY,ALL THE LITTLE PEOPLE AND,THEIR HOUSES ARE MADE
 OF,SHINY WHITE PORCELAIN CHINA,
560 DATA THE STREETS ARE SLIPPERY!,WITH EVERY STEP YOU SLIP,AND FALL. THE DELAY
COSTS,YOU A TURN
570 GOSUB 330
580 CALL CHAR(96,"55AA55AA55AA55AA"):: CALL COLOR(9,15,12):: CALL HCHAR(1,1,96,5
44):: RESTORE 600
590 GOSUB 450 :: GOSUB 350 :: RETURN
600 DATA 19,1,3,THE IMPASSABLE DESERT!,YOU MUST GO BACK OR YOU WILL,PERISH IN TH
E DEADLY SANDS
610 GOSUB 300 :: GOTO 580
620 IF Y<2 THEN 2070
630 GOSUB 300 :: IF Y>7 THEN RESTORE 650 :: TURN=TURN+1 ELSE RESTORE 640
640 DATA 2,1,5,THE TALL BUILDINGS,SPARKLE LIKE A MINE,FULL OF EMERALDS,EVERYTHIN
G IS GREEN,THE TREES AND EVEN THE WATER
650 DATA 2,1,5,THE EMERALD CITY GUARD,STOPS YOU TO MAKE SURE YOU,ARE WEARING THE
 OFFICIAL,GREEN GLASSES,THE DELAY COSTS YOU A TURN
660 GOSUB 450 :: GOSUB 350 :: RETURN
670 GOSUB 330 :: IF Y<5 THEN 680 ELSE RESTORE 700 :: GOTO 690
680 IF TIN THEN RESTORE 710 :: T=1 :: CALL TOTO(T)ELSE TURN=TURN+1 :: RESTORE 73
0
690 GOSUB 450 :: GOSUB 350 :: CALL DELSPRITE(ALL):: CALL CHARSET :: RETURN
700 DATA 4,2,5,THE FIGHTING APPLE TREES!,,WALK SOFTLY SO AS NOT TO,DISTURB THEM
OR THEY MAY,ENTANGLE YOU!
710 DATA 2,2,9,THE NASTY FIGHTING TREES,GRAB YOU!  TOTO BARKS,BUT IT DOES NO GOO
D,,LUCKILY THE TIN MAN HAS
720 DATA HIS AXE. HE CHOPS OFF,THE LIMBS THAT HAVE,ENSNARLED YOU AND SETS,YOU FR
EE!
730 DATA 2,1,9,THE FIGHTING TREES,GRAB YOU!,YOU STRUGGLE...IT'S A,TOUGH FIGHT BU
T YOU FINALLY,BREAK FREE!,,THE FIGHT EXHAUSTS YOU.
740 DATA YOU STOP TO REST AND,LOSE A TURN
750 GOSUB 330 :: IF WAY=3 THEN RESTORE 770 :: WAY=7 ELSE RESTORE 780
760 GOSUB 450 :: GOSUB 350 :: RETURN
770 DATA 4,2,4,GLINDA SITS ON HER RUBY,THRONE. SHE GIVES YOU THE,RUBY SLIPPERS A
ND SENDS YOU,BACK TO THE WIZARD
780 DATA 4,2,5,GLINDA'S RUBY PALACE,,THE GOOD SORCERESS IS AWAY,,GO ANOTHER WAY
790 GOSUB 330 :: IF Y<4 THEN RESTORE 810 ELSE IF Y>6 THEN RESTORE 830 ELSE RESTO
RE 850
800 GOSUB 450
810 DATA 2,2,9,YOU ARE WALKING DOWN THE,BACKWARDS HIGHWAY,,KEEP TRACK OF YOUR,DI
RECTION AS THIS IS A,STRANGE LAND AND EVERY
820 DATA TIME YOU TAKE A STEP THE,SIGNS SEEM TO REVERSE,THEIR DIRECTION.
```

```
830 DATA 1,1,9,THE BACKWARDS HIGHWAY TAKES,YOU ACROSS GILLIKIN COUNTRY,BUT EVERY
 TIME YOU WALK,FORWARD YOU SEEM TO,MOVE IN REVERSE,
840 DATA PERHAPS IF YOU TRIED,WALKING BACKWARDS YOU,WOULD MAKE SOME PROGRESS
850 DATA 2,1,7,A SIGNPOST POINTS IN BOTH,DIRECTIONS. IT READS:,,THE GREAT BACKWA
RDS HIGHWAY,,,VERY CONFUSING INDEED!
860 CALL KEY(0,K,S):: IF S=0 THEN 860 ELSE IF POS("NSEWMQ",CHR$(K),1)=0 THEN CAL
L SOUND(50,110,0):: GOTO 860
870 IF K=78 THEN R=R+1 ELSE IF K=83 THEN R=R-1 ELSE IF K=87 THEN C=C+1 ELSE IF K
=69 THEN C=C-1
880 IF K=77 THEN CALL MAP(R,C)ELSE IF K=81 THEN 2250
890 RETURN
900 GOSUB 330 :: IF Y<5 THEN 910 ELSE RESTORE 930 :: GOTO 920
910 IF LION THEN RESTORE 940 ELSE TURN=TURN+1 :: RESTORE 960
920 GOSUB 450 :: GOSUB 350 :: RETURN
930 DATA 2,1,5,A DARK AND CREEPY FOREST,DO YOU THINK WE'LL MEET,ANY WILD ANIMALS
?,WE MIGHT MEET SOME,LIONS AND TIGERS AND BEARS!
940 DATA 2,1,8,OH NO! THERE ARE WILD,ANIMALS ALL AROUND YOU!,THEY START TO ATTAC
K...,,JUST IN TIME THE LION,MUSTERS ENOUGH COURAGE TO
950 DATA SCARE THEM AWAY WITH HIS,MIGHTY ROAR!
960 DATA 2,1,9,WILD ANIMALS SPRING FORTH,FROM THE DARK FOREST!,,I'D TURN BACK IF
 I,WERE YOU!,
970 DATA BETTER TO LOSE A TURN,THAN TO END UP A BEAST'S,DINNER!
980 GOSUB 330 :: IF Y<3 THEN RESTORE 1000 ELSE IF Y>7 THEN TURN=TURN+1 :: GOTO 1
040 ELSE RESTORE 1030
990 GOSUB 450 :: GOSUB 350 :: RETURN
1000 DATA 2,1,6,RUGGED TERRAIN HERE,BE CAREFUL - IT MAY BE ROUGH,GOING FOR AWHIL
E,THE ROCKS AND HILLS ARE,INTERESTING BUT MAKE FOR
1010 DATA A TIRING JOURNEY
1020 DATA 2,1,5,OOPS!  YOU SLIP AND,SLIDE DOWN THE ROCKS!,BETTER REST AWHILE,YOU
 LOSE A TURN THEN DUST,YOURSELF OFF AND MOVE ON
1030 DATA 2,1,4,ROCKY GROUND,WATCH YOUR STEP,UP HILL AND DOWN HILL,YOU TRUDGE ON
WARD
1040 RESTORE 1020 :: GOSUB 450 :: FOR I=1000 TO 200 STEP -50 :: CALL SOUND(90,I,
0):: NEXT I :: CALL SOUND(250,-7,0):: GOSUB 350 :: RETURN
1050 GOSUB 330 :: IF R>14 THEN 1110
1060 IF Y<5 THEN RESTORE 1080 ELSE RESTORE 1090
1070 GOTO 1140
1080 DATA 2,1,5,A GILLIKIN VILLAGE,THE HOUSES IN THIS LAND ARE,MADE OF PURPLE BR
ICK,NO ONE IS AROUND,THEY MUST BE HIDING
1090 DATA 2,1,9,A VILLAGE,THE GILLIKINS LOOK OUT OF,THEIR WINDOWS TO SEE IF,YOU
ARE FRIEND OR FOE,,THOUGH THEY ARE RULED BY
1100 DATA THE KIND OZMA THEY MUST BE,CAUTIOUS IN THIS UNCIVILIZED,COUNTRY
1110 IF Y<5 THEN RESTORE 1120 ELSE RESTORE 1130
1120 DATA 2,1,5,A QUADLING VILLAGE,THE COTTAGES ARE ALL MADE,OF RED BRICK,RED IS
 THE FAVORITE COLOR,OF THE QUADLINGS
1130 DATA 2,1,4,A VILLAGE OF QUADLINGS,THEY ARE ALL DRESSED IN RED,THE FAVORITE
COLOR OF,GLINDA THEIR KIND RULER
1140 GOSUB 450 :: GOSUB 350 :: RETURN
1150 GOSUB 330 :: IF R<5 AND C<10 AND WAY=1 THEN 2040
1160 CALL CHAR(96,"04081828C7"):: CALL COLOR(9,15,6):: CALL HCHAR(17,1,96,256)
1170 RESTORE 1180 :: GOSUB 450 :: R=LR :: C=LC :: GOSUB 350 :: RETURN
1180 DATA 4,4,4,A LAKE IS HERE,,YOU'LL HAVE TO GO,ANOTHER WAY
1190 GOSUB 330 :: RESTORE 1200 :: GOSUB 450 :: R=LR :: C=LC :: GOSUB 350 :: RETU
RN
1200 DATA 5,3,5,A HIGH MOUNTAIN RANGE,,THE CLIFFS ARE TOO STEEP,TO CLIMB. YOU'LL
 HAVE,TO GO ANOTHER WAY
1210 GOSUB 300 :: CALL COLOR(9,2,2):: CALL HCHAR(12,1,96,416)
1220 IF TIN THEN RESTORE 1240 ELSE RESTORE 1250 :: R=LR :: C=LC
1230 GOSUB 450 :: GOSUB 350 :: RETURN
1240 DATA 1,1,4,A DEEP RAVINE! THE TIN MAN,CHOPS A TREE & LAYS THE LOG,ACROSS TH
E PIT. YOU CAN,CROSS BUT WATCH YOUR STEP!
1250 DATA 2,6,3,A DEEP RAVINE!,YOU'LL HAVE TO GO,ANOTHER WAY
1260 GOSUB 300 :: CALL CHAR(96,"55AA55AA55AA55AA"):: CALL COLOR(9,15,4):: CALL H
CHAR(1,1,96,544)
1270 IF Y<7 THEN RESTORE 1290 ELSE RESTORE 1300
```

```
1280 GOSUB 450 :: GOSUB 350 :: RETURN
1290 DATA 19,1,3,A VAST PRAIRIE LIES BEFORE,YOU. NOT A SIGN OF LIFE,FOR MILES AH
EAD
1300 DATA 19,1,3,THIS DRY LIFELESS PRAIRIE,REMINDS YOU OF KANSAS.,SURE YOU WANT
TO GO BACK?
1310 GOSUB 300 :: CALL CHAR(96,"183C3C70FFFF7E18"):: CALL COLOR(9,7,1):: CALL HC
HAR(14,1,96,352)
1320 IF Y(3 THEN RESTORE 1340 ELSE IF Y)6 THEN RESTORE 1350 ELSE RESTORE 1360 ::
 TURN=TURN+1
1330 GOSUB 450 :: GOSUB 350 :: RETURN
1340 DATA 2,1,4,THE TANTALIZING AROMA OF,THE DEADLY POPPIES MAKES,YOU DROWSY,BET
TER KEEP MOVING
1350 DATA 2,1,5,THE DEADLY POPPY FIELD,RED POPPIES AS FAR AS YOU,CAN SEE. THEIR
BEAUTY CAN,BE DECEIVING,BEWARE OF THEIR SCENT
1360 DATA 2,2,4,THE SPELL OF THE DEADLY,POPPIES OVERWHELMS YOU!,YOU FALL ASLEEP
& LOSE A,TURN.  Z-Z-Z-Z-Z-Z-Z-Z
1370 GOSUB 300 :: IF WAY=0 THEN 1470 ELSE IF WAY)4 THEN 1400
1380 RESTORE 1390 :: GOTO 1540
1390 DATA 4,3,6,THE WIZARD'S PALACE,,THIS GRAND HALL WAS BUILT,BY THE PEOPLE OF,
EMERALD CITY TO HONOR,THEIR KINDLY RULER
1400 IF WAY=5 THEN RESTORE 1420 ELSE IF WAY=6 THEN RESTORE 1430 ELSE IF WAY=7 TH
EN RESTORE 1450 ELSE IF WAY=8 THEN RESTORE 1460
1410 GOSUB 450 :: GOTO 280
1420 DATA 2,2,6,THE WIZARD SAYS SOME,MAGIC WORDS,,OZMA'S MAGIC BELT,TRANSPORTS Y
OU AND TOTO,BACK TO KANSAS IN A FLASH!
1430 DATA 2,2,8,THE WIZARD MIXES THE,MAGIC POWDER WITH A,POTION IN HIS LABORATOR
Y,,A CLOUD OF SMOKE ENGULFS,YOU AND TOTO. BEFORE YOU
1440 DATA KNOW IT YOU ARE BACK,IN KANSAS
1450 DATA 2,2,7,THE WIZARD TELLS YOU TO,TURN AROUND AND,CLICK YOUR HEELS 3 TIMES
,AND THINK OF HOME!,,YOU AND TOTO FLY,BACK TO KANSAS!
1460 DATA 4,2,8,THE WIZARD TELLS YOU,THE SECRET OF THE MAGIC CAP,,SAY HIS MAGIC
WORD 3 TIMES,AND MAKE A WISH,,NOW YOU AND TOTO,CAN GO HOME!
1470 I=INT(RND*5)+1 :: IF I=5 THEN WAY=4 ELSE WAY=I
1480 IF WAY=1 THEN RESTORE 1490 ELSE IF WAY=2 THEN RESTORE 1500 ELSE IF WAY=3 TH
EN RESTORE 1520 ELSE IF WAY=4 THEN RESTORE 1530
1490 DATA 4,2,5,THE WIZARD SAYS GO TO,PRINCESS OZMA'S PALACE,IN THE NORTHWESTERN
 LAKES,OF GILLIKAN COUNTRY,TO GET HER MAGIC BELT!
1500 DATA 4,2,7,THE WIZARD TELLS YOU OF,PROFESSOR WOGGLEBUG WHO,LIVES IN THE NOR
THEAST IN,MUNCHKIN LAND,
1510 DATA BRING BACK THE PROFESSOR'S,MAGIC POWDER!
1520 DATA 4,2,7,GLINDA THE GOOD WITCH,OF THE SOUTH HAS,THE RUBY SLIPPERS,,BRING
THEM TO THE WIZARD,SO HE CAN HELP YOU,RETURN HOME!
1530 DATA 4,2,5,THE WIZARD SAYS YOU MUST,BRING HIM THE MAGIC,GOLD CAP FROM THE C
ASTLE,OF THE WICKED WITCH,OF THE WEST TO GET HOME!
1540 GOSUB 450 :: GOSUB 350 :: RETURN
1550 GOSUB 330 :: GOSUB 2160 :: CALL CHAR(96,"FF818181FF818181"):: CALL COLOR(9,
15,9):: CALL VCHAR(1,4,96,192)
1560 IF Y(6 THEN RESTORE 1580 ELSE RESTORE 1590
1570 GOSUB 450 :: GOSUB 350 :: RETURN
1580 DATA 4,14,3,EASE ON DOWN,THE RED BRICK,ROAD!
1590 DATA 3,13,6,THE RED BRICK,ROAD WINDS,SOUTH THROUGH,THE QUADLING,COUNTRY TOW
ARDS,GLINDA'S CASTLE
1600 GOSUB 300 :: IF Y(3 THEN 1620 ELSE IF Y)6 THEN RESTORE 1670 ELSE RESTORE 16
90
1610 CALL COLOR(9,3,3):: CALL HCHAR(16,1,96,288):: GOSUB 450 :: GOSUB 350 :: RET
URN
1620 CALL CHAR(96,"183C3C7EFFFF7E18"):: CALL COLOR(9,16,1):: CALL HCHAR(14,1,96,
352):: RESTORE 1660 :: GOSUB 450
1630 CALL KEY(0,K,S)
1640 I=INT(RND*10)+7 :: IF I=11 OR I=12 THEN 1640 ELSE CALL COLOR(9,I,1)
1650 IF S=0 THEN 1630 ELSE IF POS("NSEWMQ",CHR$(K),1)=0 THEN 1630 ELSE GOSUB 400
 :: RETURN
1660 DATA 1,2,5,A FIELD OF BEAUTIFUL,RAINBOW FLOWERS,THEY CAN'T DECIDE WHAT,COLO
R TO BE SO THEY CHANGE,COLORS BEFORE YOUR EYES!
```

```
1670 DATA 2,2,7,A GRASSY MEADOW LIES,BEFORE YOU,,THE BLUE SKIES AND SINGING,BIRD
S MAKE YOU THINK -,'OZ MAY BE STRANGE
1680 DATA BUT IT SURE IS BEAUTIFUL!'
1690 DATA 5,3,3,YOU ARE RUNNING THROUGH,A LARGE MEADOW,MAKING GOOD PROGRESS!
1700 GOSUB 330 :: IF Y<6 THEN RESTORE 1730 ELSE RESTORE 1740
1710 GOSUB 450 :: FOR I=1 TO 12 :: CALL SCREEN(16):: CALL SOUND(-500,-7,0):: CAL
L SCREEN(2):: NEXT I
1720 CALL SCREEN(SCR):: GOSUB 350 :: TURN=TURN+1 :: RETURN
1730 DATA 4,5,6,EARTHQUAKE TREMORS!,,YOU ARE A BIT SHAKEN,SO YOU STOP TO GET,YOU
R BEARINGS,AND LOSE A TURN
1740 DATA 4,5,5,THUNDERSTORMS!,,YOU LOSE A TURN,WAITING OUT THE RAIN,UNDER A BIG
 TREE
1750 GOSUB 330 :: RESTORE 1780 :: CALL COLOR(9,15,5):: A$="2020204040408080" ::
B$="4040808020202040" :: CALL CHAR(96,A$)
1760 CALL VCHAR(1,4,96,192):: GOSUB 450
1770 CALL CHAR(96,B$):: CALL SOUND(-750,-6,10):: CALL KEY(0,K,S):: CALL CHAR(96,
A$):: IF S=0 THEN 1770 ELSE C=C+2 :: RETURN
1780 DATA 3,12,6,THE UP-AND-DOWN,WATERFALL CARRIES,YOU UP AND,OVER THE HIGH,MOUN
TAINS TO,THE EAST
1790 GOSUB 330 :: RESTORE 1800 :: GOSUB 450 :: GOSUB 350 :: RETURN
1800 DATA 3,2,7,MIST VALLEY,,THE FOG IS SO DENSE HERE,YOU CAN'T TELL WHICH WAY,T
O GO.  YOU MUST KEEP,WANDERING UNTIL YOU FIND,A WAY OUT
1810 GOSUB 300 :: RESTORE 1850 :: GOSUB 450
1820 CALL KEY(0,K,S):: IF S=0 THEN 1820 ELSE IF K=13 THEN 1830 ELSE IF POS("NSEW
MQ",CHR$(K),1)=0 THEN 1820 ELSE GOSUB 400 :: RETURN
1830 R=INT(RND*24)+1 :: C=INT(RND*32)+1 :: PRINT "HERE GOES...": : :: CALL COLOR
(9,2,2)
1840 FOR I=1 TO 24 :: PRINT RPT$(CHR$(96),28):: CALL SOUND(-300,1000-(I*30),0)::
 NEXT I :: RETURN
1850 DATA 2,1,11,YOU ARE AT THE ENTRANCE TO,A MAGIC WISHWAY TUNNEL THAT,CAN TRAN
SPORT YOU TO ANY,PLACE IN OZ. YOU CAN WISH
1860 DATA TO BE IN THE LAND OF YOUR,CHOICE BUT SOMETIMES THE,MAGIC WILL BACKFIRE
 AND,TRANSPORT YOU TO THE WRONG,PLACE.  PRESS 'ENTER' TO
1870 DATA ENTER THE WISHWAY OR CHOOSE,A DIRECTION TO MOVE ON
1880 GOSUB 300 :: IF C>19 THEN 1950 ELSE IF WAY=4 THEN GOSUB 2140 :: GOTO 1910 E
LSE RESTORE 1900
1890 GOSUB 450 :: GOSUB 350 :: RETURN
1900 DATA 4,3,5,THE AWFUL CASTLE OF THE,WICKED WITCH!,,I'D TURN BACK,IF I WERE Y
OU!
1910 DISPLAY AT(3,12):"THE CASTLE OF" :: DISPLAY AT(5,12):"THE WICKED WITCH" ::
DISPLAY AT(8,14):"SHE HAS YOU"
1920 DISPLAY AT(10,14):"TRAPPED!!"
1930 CALL WAIT :: TURN=TURN+1 :: IF Y<8 THEN Y=Y+1 :: CALL SOUND(150,110,0):: GO
TO 1930
1940 CALL VCHAR(1,14,32,456):: CALL TOTO(T):: GOSUB 350 :: CALL DELSPRITE(ALL)::
 CALL CLEAR :: CALL CHARSET :: WAY=8 :: RETURN
1950 IF WAY=2 THEN RESTORE 1960 :: WAY=6 :: GOSUB 450 :: GOSUB 350 :: RETURN ELS
E RESTORE 1980 :: GOTO 1890
1960 DATA 3,2,7,PROFESSOR WOGGLEBUG GIVES,YOU A PINCH OF MAGIC,POWDER BUT WON'T
DIVULGE,HIS TRADE SECRETS!,
1970 DATA GO ASK THE WIZARD HOW,TO USE IT
1980 DATA 6,4,4,PROFESSOR WOGGLEBUG'S,COTTAGE,,NOBODY HOME!
1990 GOSUB 300 :: GOSUB 2160 :: CALL CHAR(96,"FF818181FF818181"):: CALL COLOR(9,
16,12):: CALL VCHAR(1,4,96,192)
2000 IF Y<6 THEN RESTORE 2020 :: GOSUB 450 :: CALL WIZ ELSE RESTORE 2030 :: GOSU
B 450
2010 GOSUB 350 :: RETURN
2020 DATA 4,14,3,JUST FOLLOW,THE YELLOW,BRICK ROAD!
2030 DATA 2,14,7,THE YELLOW,BRICK ROAD,WINDS ACROSS,MUNCHKIN LAND,AND THROUGH,TH
E WINKIE,COUNTRY
2040 GOSUB 330 :: RESTORE 2050 :: GOSUB 450 :: CALL WAIT :: R=3 :: C=10 :: WAY=5
 :: RETURN
2050 DATA 3,2,6,PRINCESS OZMA LOWERS THE,BRIDGE TO HER CRYSTAL,PALACE. SHE LETS
YOU BORROW,HER MAGIC BELT AND PUTS YOU
2060 DATA ON THE BACKWARD HIGHWAY,TO EMERALD CITY
```

```
2070 A$="SURRENDER DOROTHY" :: GOSUB 300 :: RESTORE 2080 :: GOSUB 450
2080 DATA 16,1,3,DOROTHY IT LOOKS AS THOUGH,THE WITCH HAS FOLLOWED YOU!,WHAT EVE
R WILL YOU DO NOW?
2090 CALL CHAR(96,"0000000103000000000F0FFF00103070F4040E0F0F8607040E0D0FFC0C0C0C
0C0")
2100 CALL SPRITE(#1,96,2,21,1,0,10):: CALL MAGNIFY(3)
2110 CALL POSITION(#1,DR,DC):: IF DC<62 THEN 2110
2120 FOR I=1 TO 17 :: CALL HCHAR(4,7+I,ASC(SEG$(A$,I,1))):: FOR D=1 TO 30 :: NEX
T D :: NEXT I
2130 CALL POSITION(#1,DR,DC):: IF DC<250 THEN 2130 ELSE CALL DELSPRITE(#1):: GOS
UB 350 :: RETURN
2140 CALL CLEAR :: FOR I=1 TO 12 :: CALL COLOR(I,2,4):: NEXT I :: FOR I=1 TO 54
:: CALL CHAR(I+89,W$(I)):: NEXT I
2150 CALL COLOR(13,7,4,14,2,13):: FOR I=1 TO 19 :: DISPLAY AT(I+3,3):WD$(I):: NE
XT I :: RETURN
2160 IF TIN=0 AND Y=1 THEN TIN=1 :: RESTORE 2200 :: GOTO 2190
2170 IF CROW=0 AND Y=2 THEN CROW=1 :: RESTORE 2210 :: GOTO 2190
2180 IF LION=0 AND Y=3 THEN LION=1 :: RESTORE 2230 ELSE RETURN
2190 GOSUB 450 :: DISPLAY AT(24,7):"PRESS ANY KEY" :: CALL WAIT :: CALL CLEAR ::
 RETURN
2200 DATA 3,3,7,YOU HAVE JUST MET THE,TIN WOODSMAN,,WITH HIS TRUSTY,RUSTY AXE HE
 WOULD,MAKE A HELPFUL,TRAVELING COMPANION
2210 DATA 3,3,8,YOU HAVE JUST MET THE,SCARECROW. HE SUGGESTS YOU,VISIT THE WIZAR
D IN THE,EMERALD CITY TO FIND OUT,HOW TO GET HOME
2220 DATA ,THE SCARECROW WILL,ACCOMPANY YOU ON YOUR TRIP
2230 DATA 3,2,7,THE COWARDLY LION,,HE IS COWARDLY NO MORE,SINCE THE WIZARD GAVE
HIM,COURAGE.  HE WILL ESCORT
2240 DATA YOU TO HELP AVOID,DANGER AND WILD ANIMALS
2250 CALL CLEAR :: PRINT "PRESS":"Q TO QUIT GAME":"R TO RESTART ADVENTURE":"C TO
 CONTINUE"
2260 CALL KEY(0,K,S):: IF S=0 THEN 2260
2270 IF K=67 THEN RETURN
2280 IF K=82 THEN TURN,WAY,LION,TIN,CROW=0 :: R=INT(RND*22)+2 :: C=INT(RND*30)+2
 :: RETURN
2290 IF K<>81 THEN 2260
2300 CALL CLEAR :: CALL SAY("GOODBYE"):: STOP
2310 SUB MAP(R,C)
2320 CALL CLEAR :: CALL SCREEN(4)
2330 CALL CHAR(128,"55AA55AA55AA55AA"):: CALL COLOR(5,2,1,6,2,1,7,2,1,8,2,1)
2340 CALL COLOR(9,13,13,10,6,6,11,7,7,12,12,12,13,14,5)
2350 CALL HCHAR(1,1,128,384):: CALL HCHAR(13,1,112,384)
2360 RP=24 :: R1=1 :: FOR C1=1 TO 11 :: CALL VCHAR(R1,C1,120,RP):: RP=RP-2 :: R1
=R1+1 :: NEXT C1
2370 RP=4 :: R1=11 :: FOR C1=20 TO 30 :: CALL VCHAR(R1,C1,104,RP):: RP=RP+2 :: R
1=R1-1 :: NEXT C1
2380 CALL VCHAR(1,31,104,24):: CALL VCHAR(1,32,104,24):: FOR R1=11 TO 14 :: CALL
 HCHAR(R1,12,96,8):: NEXT R1
2390 RESTORE 2400 :: FOR I=1 TO 6 :: READ R1,C1,B$ :: DISPLAY AT(R1,C1)SIZE(LEN(
B$)):B$ :: NEXT I
2400 DATA 6,10,GILLIKIN,18,10,QUADLING,12,1,WINKIE,12,20,MUNCHKIN,12,11,EMERALD,
13,12,CITY,
2410 CALL CHAR(136,"0044442810284444",137,"0"):: CALL COLOR(14,9,16)
2420 CALL KEY(0,K,S):: CALL HCHAR(R,C,136):: FOR D=1 TO 70 :: NEXT D
2430 CALL HCHAR(R,C,137):: FOR D=1 TO 30 :: NEXT D :: IF S=0 THEN 2420
2440 SUBEND
2450 SUB TITLE
2460 CALL CHAR(96,"FF8080808080FF",97,"FF000000000000FF",104,"FFFFFFFFFFFFFFFF
"):: CALL COLOR(9,16,12,10,13,13)
2470 CALL CLEAR :: CALL SCREEN(12)
2480 FOR I=1 TO 31 STEP 2 :: CALL VCHAR(1,I,96,24):: CALL VCHAR(1,I+1,97,24):: N
EXT I
2490 DISPLAY AT(2,8)SIZE(12):"ADVENTURE IN"
2500 CALL HCHAR(5,8,104,7):: CALL HCHAR(6,7,104,9):: CALL HCHAR(19,7,104,9):: CA
LL HCHAR(20,8,104,7)
```

```
2510 CALL VCHAR(7,6,104,12):: CALL VCHAR(7,7,104,12):: CALL VCHAR(7,15,104,12)::
 CALL VCHAR(7,16,104,12)
2520 CALL HCHAR(5,19,104,9):: CALL HCHAR(6,19,104,9):: CALL HCHAR(19,19,104,9)::
 CALL HCHAR(20,19,104,9)
2530 C=25.5 :: FOR R=7 TO 18 :: CALL HCHAR(R,INT(C),104,2):: C=C-.5 :: NEXT R
2540 CALL WIZ :: DISPLAY AT(23,8)SIZE(13):"PRESS ANY KEY"
2550 CALL WAIT
2560 SUBEND
2570 SUB CYCLONE
2580 CALL CLEAR :: CALL SCREEN(15)
2590 DISPLAY AT(2,1):"DOROTHY! A CYCLONE  LIFTS UP" :: DISPLAY AT(4,1):"YOUR HOU
SE AND TAKES YOU AND"
2600 DISPLAY AT(6,1):"TOTO FAR, FAR AWAY TO THE" :: DISPLAY AT(8,1):"LAND OF OZ.
.." :: FOR D=1 TO 300 :: NEXT D
2610 CALL CHAR(96,"FF7F3F1F0F0703010100000000000000FFFEFEFCFCFCF8F8F8F8F0F070703
010")
2620 CALL CHAR(100,"0000000000030F3FFF7F636363637F7F7F00000000C0F0FCFFFEC6C6C6C6FE
FEFE")
2630 CALL SPRITE(#1,96,2,160,192,0,-10):: CALL MAGNIFY(4)
2640 CALL COLOR(9,16,1):: CALL HCHAR(23,8,100):: CALL HCHAR(24,8,101):: CALL HCH
AR(23,9,102):: CALL HCHAR(24,9,103)
2650 CALL SOUND(3630,110,6,-6,10)
2660 CALL COINC(#1,160,46,4,A):: IF A=0 THEN 2660 ELSE CALL HCHAR(23,8,32,2):: C
ALL HCHAR(24,8,32,2):: CALL SOUND(60,660,6,770,6)
2670 CALL DELSPRITE(#1)
2680 CALL CLEAR :: CALL SCREEN(2):: A$=RPT$("F",16)
2690 FOR I=9 TO 14 :: CALL COLOR(I,2,2):: NEXT I
2700 FOR I=96 TO 128 STEP 8 :: CALL CHAR(I,A$):: NEXT I
2710 CC=96 :: FOR I=1 TO 24 :: CALL SOUND(-100,300+25*I,3):: CALL HCHAR(I,1,CC,3
2):: CC=CC+8 :: IF CC>128 THEN CC=96
2720 NEXT I
2730 CALL CHAR(136,"0000000000030F3FFF7F636363637F7F7F00000000C0F0FCFFFEC6C6C6C6FE
FEFE")
2740 CALL SPRITE(#1,136,16,1,76,15,5):: CALL MAGNIFY(3):: F=1000
2750 F=F-20 :: CALL SOUND(-100,F,0):: CALL POSITION(#1,B,B1):: IF B<165 THEN 275
0 ELSE CALL MOTION(#1,0,0):: CALL SOUND(100,110,3,-7,4)
2760 CALL SCREEN(16):: CALL COLOR(9,7,7,10,12,12,11,13,13,12,5,5,13,14,14):: FOR
 D=1 TO 125 :: NEXT D
2770 SUBEND
2780 SUB RAINBOW
2790 CALL SOUND(922,196,6,233,6,311,0):: CALL SOUND(922,392,6,466,6,622,0):: CAL
L SOUND(461,294,6,466,6,587,0)
2800 CALL SOUND(230,294,6,392,6,466,0):: CALL SOUND(230,294,6,440,6,523,0):: CAL
L SOUND(461,294,6,466,6,587,0)
2810 CALL SOUND(461,277,6,523,6,622,0):: CALL SOUND(922,208,6,262,6,311,0):: CAL
L SOUND(922,156,6,392,6,523,0)
2820 CALL SOUND(922,117,8,294,6,466,0):: CALL SOUND(500,40000,30)
2830 SUBEND
2840 SUB WIZ
2850 A=0 :: D=166
2860 CALL SOUND(499,784,A):: CALL SOUND(333,1047,A):: CALL SOUND(D,698,A):: CALL
 SOUND(333,659,A):: CALL SOUND(D,698,A):: CALL SOUND(D,784,A)
2870 CALL SOUND(499,1047,A):: CALL SOUND(D,40000,30):: CALL SOUND(D,196,A):: CAL
L SOUND(D,220,A):: CALL SOUND(D,196,A):: CALL SOUND(D,175,A)
2880 CALL SOUND(D,165,A):: CALL SOUND(D,175,A):: CALL SOUND(D,147,A):: CALL SOUN
D(666,131,A)
2890 SUBEND
2900 SUB INSTRUCT
2910 CALL CLEAR :: DISPLAY AT(2,7):"ADVENTURE IN OZ" :: DISPLAY AT(4,2):RPT$("-"
,26):: DISPLAY AT(6,2):"HOW MANY TURNS WILL IT"
2920 DISPLAY AT(7,2):"TAKE YOU TO GET BACK HOME?"
2930 DISPLAY AT(9,2):"USE THESE KEY COMMANDS:"
2940 DISPLAY AT(11,4):"N = MOVE NORTH" :: DISPLAY AT(13,4):"S = MOVE SOUTH" :: D
ISPLAY AT(15,4):"E = MOVE EAST"
```

```
2950 DISPLAY AT(17,4):"W = MOVE WEST" :: DISPLAY AT(19,4):"M = MAP (DRAW LOCATIO
N)" :: DISPLAY AT(21,4):"Q = QUIT (END GAME)"
2960 DISPLAY AT(24,2):"(PRESS A KEY TO CONTINUE)"
2970 CALL WAIT :: CALL CLEAR
2980 DISPLAY AT(4,2):"MAYBE THE WIZARD CAN HELP" :: DISPLAY AT(6,2):"YOU GET TO
KANSAS!"
2990 DISPLAY AT(8,2):"YOU MAY LOSE A TURN IF YOU" :: DISPLAY AT(10,2):"TAKE A TR
EACHEROUS ROUTE"
3000 DISPLAY AT(12,2):"OR STOP TOO LONG TO READ" :: DISPLAY AT(14,2):"THE MAP, A
ND REMEMBER THAT" :: DISPLAY AT(16,2):"THIS LAND IS FULL"
3010 DISPLAY AT(18,2):"OF SURPRISES!  GOOD LUCK!" :: DISPLAY AT(22,8):"(PRESS A
KEY)"
3020 CALL WAIT :: CALL CLEAR
3030 SUBEND
3040 SUB TOTO(T)
3050 CALL MAGNIFY(4):: A$=RPT$("0",10)
3060 CALL CHAR(40,"0000003ØD8FC3F1F1F0F07040404080ØØC06Ø3Ø3Ø6ØCFCFEFEFEFC0808040
8")
3070 CALL SPRITE(#3,40,2,162,225)
3080 CALL CHAR(48,"0000003ØD8FC3F1F1F0F07080810100018ØC06Ø6ØC0CFCFEFEFEFC0804040
2")
3090 CALL MOTION(#3,0,-15):: FOR I=1 TO 25 :: CALL PATTERN(#3,48):: FOR D=1 TO 1
0 :: NEXT D :: CALL PATTERN(#3,40)
3100 FOR D=1 TO 10 :: NEXT D :: NEXT I :: CALL MOTION(#3,0,0):: IF T THEN T=0 ::
 SUBEXIT
3110 CALL CHAR(48,"0000003ØD8FC3F1F1F0F07040404080ØØC06Ø3Ø3Ø6ØCFCFEFEFEFC0808040
8")
3120 CALL PATTERN(#3,48)
3130 CALL CHAR(48,"00000008687078ØF5F4F07040404080ØØØØØØØØ8Ø6ØC0CFEFEFEFC0808040
8")
3140 CALL SAY("#")
3150 CALL CHAR(52,RPT$(A$,4)&"00000003"):: CALL SPRITE(#4,52,11,174,41)
3160 CALL CHAR(52,RPT$(A$,4)&"000006")
3170 CALL CHAR(52,RPT$(A$,4)&"00020706")
3180 CALL CHAR(52,RPT$(A$,4)&"00061F07")
3190 CALL CHAR(52,RPT$(A$,4)&"061F3F1F")
3200 CALL CHAR(52,A$&"000000F7"&A$&A$&"00073F3F1F")
3210 CALL CHAR(52,A$&"3F7FFFFF"&A$&A$&"07373FFF3F")
3220 CALL CHAR(48,"0000003ØD8FC3F1F1F0F07040404080ØØC06Ø3Ø3Ø6ØCFCFEFEFEFC0808040
8")
3230 FOR I=4 TO 22 :: CALL HCHAR(I,1,32,14):: CALL SOUND(-300,600-(20*I),I,-5,I)
:: NEXT I :: DISPLAY AT(5,3):"TOTO HAS SAVED THE DAY!"
3240 DISPLAY AT(7,3):"HE LIQUIDATED THE WITCH!" :: DISPLAY AT(9,3):"NOW THAT SHE
 HAS MELTED"
3250 DISPLAY AT(11,3):"YOU CAN TAKE THE GOLDEN" :: DISPLAY AT(13,3):"CAP AND HEA
D FOR THE" :: DISPLAY AT(15,7):"EMERALD CITY!"
3260 SUBEND
3270 SUB WAIT
3280 CALL KEY(0,K,S):: A=RND :: IF S=0 THEN 3280
3290 SUBEND
```

```
100 REM *OZFILE1*  TI EXTENDED BASIC
110 REM CREATES OZDATA1 DISK FILE FOR USE WITH ADVENTURE IN OZ GAME
120 REM DATA FILE SHOULD BE SAVED ON SAME DISK AS GAME
130 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
140 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
150 DIM R$(24)
160 R$(1)="DCCCCCCCCCCCCCCCCCCCCCCCCCCCCDD"
170 R$(2)="DNJJJJLLLJJTJJJJVJJJJJHJJFFJOOOD"
180 R$(3)="DONTJJLZLHJJJMVVVVVMKKHFFJTOOOOD"
190 R$(4)="DOOOTJLLLHJJJMVVVVUMKKHILTOOOOOD"
200 R$(5)="DOPOSTJJFHFJUMVVVVVMIHHILSSSWOOD"
210 R$(6)="DOPOSAIJFHFJJMJVVJIMIHIISNSSOOXD"
220 R$(7)="DOOWSASIJHHHHHHJJIIHHHISSNSSAOYD"
230 R$(8)="DOOOSAAAIJMJJJHHHHHHIISSSNSSAOYD"
240 R$(9)="DYYYSSSASTUMJJJJJHIIISSSSSSAAYYD"
250 R$(10)="DYNYYYSASSJJMJJJJHIISSSPSSSAOYOD"
260 R$(11)="DYNNSYPASSSEEEEEERESSSPPPSSAOYOD"
270 R$(12)="DYXNSYYAYSSEWEQQRRESSPPPPSSAOYOD"
280 R$(13)="DONNSPSAYYYYYYQQYYYYYYYYPYYYAYYOD"
290 R$(14)="DOOOPPSASSSEEEEREYESSSPPPOOAOOOD"
300 R$(15)="DOOAAAAASSJJJJJRJJIJSSSPOOAAOOOD"
310 R$(16)="DOAASSSSSMJLLJJRJJIJTOOOOAOOSSD"
320 R$(17)="DOOOSSSSUMKKJJFRFJIIJTONOAAOWSSD"
330 R$(18)="DOOWSSSJJMKKFFFFFFFIKKKSSASOOSSD"
340 R$(19)="DOOSSSJKKMJJJJFRFJIIKKKMSSSNOOOD"
350 R$(20)="DOOOSMJKKJJJJJJRJJIIIJJMJSSNNPPD"
360 R$(21)="DNOOUMJJTTTTJJBBBBJIIJUMJJSSNOOD"
370 R$(22)="DNOKKMJITILLJGRRBJJLLJJMKKTSSOOD"
380 R$(23)="DNJKKJIIIIJJJJJJJJJJJJJJKKJJSOOD"
390 R$(24)=R$(1)
400 CALL CLEAR :: PRINT "PLACE DISK IN DRIVE 1 THEN"
410 INPUT "PRESS ENTER TO CONTINUE":Y$
420 PRINT :"SAVING FILE..."
430 OPEN #1:"DSK1.OZDATA1",OUTPUT,INTERNAL,FIXED 40
440 FOR I=1 TO 24 :: PRINT #1:R$(I):: NEXT I
450 CLOSE #1 :: CALL CLEAR :: STOP
```

```
100 REM *OZFILE2* TI EXTENDED BASIC
110 REM CREATES OZDATA2 DISK FILE FOR USE WITH ADVENTURE IN OZ GAME
120 REM DATA FILE SHOULD BE SAVED ON SAME DISK AS GAME
130 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
140 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
150 DIM W$(54),WD$(19)
160 FOR I=1 TO 19 :: WD$(I)="" :: FOR J=1 TO 8 :: READ N :: WD$(I)=WD$(I)&CHR$(N
)):: NEXT J :: NEXT I
170 RESTORE 440 :: FOR I=1 TO 54 :: READ W$(I):: NEXT I
180 CALL CLEAR :: PRINT "PLACE DISK IN DRIVE 1 THEN"
190 INPUT "PRESS ENTER TO CONTINUE":Y$
200 PRINT :"SAVING FILE..."
210 OPEN #1:"DSK1.OZDATA2",OUTPUT,INTERNAL,FIXED 20
220 FOR I=1 TO 19 :: PRINT #1:WD$(I):: NEXT I
230 FOR I=1 TO 54 :: PRINT #1:W$(I):: NEXT I
240 CLOSE #1 :: CALL CLEAR :: STOP
250 DATA 32,32,32,91,32,32,32,32
260 DATA 32,32,32,92,93,32,32,32
270 DATA 32,32,94,90,95,96,32,32
280 DATA 32,32,97,140,141,98,32,32
290 DATA 32,99,100,142,143,101,102,32
300 DATA 32,103,90,90,90,90,104,32
310 DATA 32,105,106,90,90,107,106,109
320 DATA 110,111,112,90,90,113,114,115
330 DATA 32,116,117,90,90,118,119,120
```

```
340 DATA 32,121,90,90,90,90,122,32
350 DATA 32,32,90,90,90,90,32,32
360 DATA 32,32,90,90,90,90,32,32
370 DATA 32,32,90,90,90,90,32,32
380 DATA 32,32,90,90,90,90,32,32
390 DATA 32,32,90,90,90,90,32,32
400 DATA 32,32,90,90,90,90,32,32
410 DATA 32,32,90,90,90,90,32,32
420 DATA 32,127,128,129,130,32,32,32
430 DATA 32,131,132,133,134,135,32,32
440 DATA FFFFFFFFFFFFFFFF,040E0E0E0E1F1F1F,3F3F3F7F7F7F7F7F,808080C0C0C0E0E,0000
000001010103,F0F0F8F8FCFCFEFF
450 DATA 0000000000000F,FFFF3F1F,F8FC06,0000000000010303,001F7FFFFFFFFFF
460 DATA 00F8FFFFFFFFFFFF,0000000808080C,07070F0F1F1F1F3F,E0E0F0F0F0F8F8FC
470 DATA 7F7F7F7EFEFEFEFC,7F7F3F3F3F3F3F1F,FEFEFCFCFCFCFCF8,FEFE7F7F3F1F1F0F,000
000008080C0C
480 DATA 0000010101010101,FCFCF8F8F8F8F8F8,1F1F1F1F0F0F0F0F,F8F8F8F8F0F0F0F,0F07
07030303070F
490 DATA E0E0F0F0F0F0E0E,FCFCFCFCFC7C7C7C,0F0707070F1F7FFF,F0E0E0E0E0F8FFFF,0F1F
1F3F7F7FFEFC
500 DATA E0C0C0808,7E7E3F3F3F3F1E,F8F0E,0,0,0,0,0
510 DATA 0F0F0F0F0F0F0F0F,C0C0C0C0C0C0C0C0,FCFCFCFCFCFCFCFC,0000071F7F7F7F7F
520 DATA 0F7FFFFFFFFFFFFF,C0C0C0C0C0C0C0C0,FFFFFFFFFFFFFFFF
530 DATA 000080E0E0F8F8F8,0
540 DATA 0,0,0,FFFFE38099928242,FFE2010119110101,43C0E3F7F3F8FCFF,820387E7CF1F3F
FF
```

RAINBOW

This music program is designed to be used with the *Adventure in Oz* game. If it is used in this way, it should be saved on the same disk as the game with the filename "RAINBOW". However, the program has been written in TI BASIC so that those with just the console and a cassette recorder can at least hear the song, even though they are not able to use the Oz game. Thus, it may be typed in and used as a completely independent program.

```
100 REM *RAINBOW* TI BASIC
110 REM FOR USE WITH ADVENTURE IN OZ GAME
120 REM FROM PROGRAMS FOR THE TI HOME COMPUTER
130 REM COPYRIGHT (C) 1983 BY STEVE DAVIS
140 CALL SCREEN(2)
150 CALL CLEAR
160 FOR I=9 TO 14
170 CALL COLOR(I,2,2)
180 NEXT I
190 CC=96
200 FOR I=1 TO 24
210 CALL HCHAR(I,1,CC,32)
220 CC=CC+8
230 IF CC<129 THEN 250
240 CC=96
250 NEXT I
260 CALL COLOR(9,7,7)
270 CALL COLOR(10,12,12)
280 CALL COLOR(11,13,13)
290 CALL COLOR(12,5,5)
300 CALL COLOR(13,14,14)
```

```
310 GOSUB 410
320 CALL SOUND(230,40000,30)
330 GOSUB 410
340 CALL SOUND(115,40000,30)
350 GOSUB 720
360 CALL SOUND(10,40000,30)
370 GOSUB 410
380 CALL SOUND(1000,40000,30)
390 CALL CLEAR
400 STOP
410 A=0
420 B=6
430 C=9
440 CALL SOUND(922,196,B,233,B,311,A)
450 CALL SOUND(922,392,B,466,B,622,A)
460 CALL SOUND(461,294,B,466,B,587,A)
470 CALL SOUND(230,294,B,392,B,466,A)
480 CALL SOUND(230,294,B,440,B,523,A)
490 CALL SOUND(461,294,B,466,B,587,A)
500 CALL SOUND(461,277,B,523,B,622,A)
510 CALL SOUND(922,208,B,262,B,311,A)
520 CALL SOUND(461,156,B,392,B,523,A)
530 CALL SOUND(461,156,B,370,B,523,A)
540 CALL SOUND(461,196,B,349,B,466,A)
550 CALL SOUND(461,196,B,311,B,466,A)
560 CALL SOUND(461,175,B,294,B,466,A)
570 CALL SOUND(461,165,B,277,B,466,A)
580 CALL SOUND(922,156,B,208,B,262,A)
590 CALL SOUND(922,175,B,311,B,415,A)
600 CALL SOUND(461,233,B,311,B,392,A)
610 CALL SOUND(230,233,B,262,B,311,A)
620 CALL SOUND(230,233,B,294,B,349,A)
630 CALL SOUND(461,233,B,330,B,392,A)
640 CALL SOUND(461,233,B,277,B,415,A)
650 CALL SOUND(461,220,B,294,B,349,A)
660 CALL SOUND(230,220,B,247,B,294,A)
670 CALL SOUND(230,220,B,262,B,311,A)
680 CALL SOUND(461,208,B,294,B,349,A)
690 CALL SOUND(461,208,B,294,B,392,A)
700 CALL SOUND(1383,196,B,233,B,311,A)
710 RETURN
720 CALL SOUND(230,466,A)
730 CALL SOUND(230,156,C,311,C,392,A)
740 CALL SOUND(230,156,C,311,C,466,A)
750 CALL SOUND(230,156,C,233,C,392,A)
760 CALL SOUND(230,156,C,233,C,466,A)
770 CALL SOUND(230,156,C,262,C,392,A)
780 CALL SOUND(230,156,C,262,C,466,A)
790 CALL SOUND(230,156,C,233,C,392,A)
800 CALL SOUND(230,156,C,233,C,466,A)
810 CALL SOUND(230,262,C,311,C,415,A)
820 CALL SOUND(230,262,C,311,C,466,A)
830 CALL SOUND(230,262,C,311,C,415,A)
840 CALL SOUND(230,262,C,311,C,466,A)
850 CALL SOUND(230,233,C,294,C,415,A)
860 CALL SOUND(230,233,C,294,C,466,A)
870 CALL SOUND(230,233,C,294,C,415,A)
880 CALL SOUND(230,233,C,294,C,466,A)
890 CALL SOUND(922,156,4,392,4,523,A)
900 CALL SOUND(1383,311,4,392,4,523,A)
910 CALL SOUND(115,40000,30)
920 CALL SOUND(230,466,A)
930 CALL SOUND(230,156,C,311,C,392,A)
940 CALL SOUND(230,156,C,311,C,466,A)
950 CALL SOUND(230,156,C,233,C,392,A)
```

```
960 CALL SOUND(230,156,C,233,C,466,A)
970 CALL SOUND(230,156,C,262,C,392,A)
980 CALL SOUND(230,156,C,262,C,466,A)
990 CALL SOUND(230,156,C,233,C,392,A)
1000 CALL SOUND(230,156,C,233,C,466,A)
1010 CALL SOUND(230,262,C,311,C,440,A)
1020 CALL SOUND(230,262,C,311,C,523,A)
1030 CALL SOUND(230,262,C,311,C,440,A)
1040 CALL SOUND(230,262,C,311,C,523,A)
1050 CALL SOUND(230,185,C,311,C,440,A)
1060 CALL SOUND(230,185,C,311,C,523,A)
1070 CALL SOUND(230,185,C,311,C,440,A)
1080 CALL SOUND(230,185,C,311,C,523,A)
1090 CALL SOUND(922,349,4,466,4,587,A)
1100 CALL SOUND(922,311,4,370,4,587,A)
1110 CALL SOUND(922,349,4,523,4,698,A)
1120 CALL SOUND(922,294,4,370,4,523,A)
1130 RETURN
```

At last, here is a book of programs written especially for users of the popular Texas Instruments 99/4 and 99/4A Home Computers. Fifty unique programs in TI BASIC and TI Extended BASIC are presented in an easy-to-read format, ready for you to type into your computer and use. Most of the programs are listed in TI BASIC and may be used with just the console and a cassette recorder, while others take advantage of the special capabilities of optional TI modules and peripheral equipment, such as TI Extended BASIC, Mini-Memory, Terminal Emulator II, the TI Speech Synthesizer, a printer or the disk system. A wide variety of programs for home applications are featured, including games and entertainment, educational programs, programming aids and more. Best of all, this volume offers programs that highlight the many unique features of the TI Home Computer, such as color, graphics, music, sound or speech. Whether you are a beginner or an experienced TI Home Computer user, you will find this collection to be an entertaining and useful addition to your program library.

$14.95 US

ISBN 0-911061-00-2