How To Choose An Income Tax Program

COMPUTE!

$2.95
December
1985
Issue 67
Vol. 7, No. 12
$3.75 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

SpeedScript Revisited
Enhancements
To _SpeedScript 3.0_

Balloon Crazy
Exciting Arcade Game
For Commodore 64/128,
Atari, Apple,
IBM PC/PCjr, TI

Atari Fine Scrolling
Secrets Of Advanced
Atari Graphics

Commodore 128 Video
Design Your Own
Custom Characters

Million-Color Palette
For PC & PCjr
Create Extra Colors
Without Extra Hardware

Apple Disk Booster
Increase Disk Capacity
With This Short Program

COMPUTE!

DECEMBER 1985
VOLUME 7
NUMBER 12
ISSUE 67

FEATURES

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

TOLL FREE Subscription Order Line
800-334-0868 (In NC 919-275-9809)

GUIDE TO ARTICLES AND PROGRAMS

•
•
AT/64/128/AP/PC/PCjr/TI
64/128/VIC/+4/16/P/AT/AP/PC/PCjr/TI

AP Apple, **Mac** Macintosh, **AT** Atari, **ST,** Atari ST, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **AM** Amiga. *General interest.

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019
Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. POSTMASTER: Send address changes to: **COMPUTE!** Magazine, P.O. Box 10955, Des Moines, IA 50950. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X.

Earth will be destroyed in 12 minutes to make way for a hyperspace bypass.

Should you hitchhike into the next galaxy? Or stay and drink beer?

Slip the disk in your computer and suddenly you are Arthur Dent, the dubious hero of THE HITCHHIKER'S GUIDE TO THE GALAXY, a side-splitting masterwork of interactive fiction by novelist Douglas Adams and Infocom's Steve Meretzky. And every decision you make will shape the story's outcome. Suppose for instance you decide to linger in the pub. You simply type, in plain English:

>DRINK THE BEER

And the story responds:

YOU GET DRUNK AND HAVE A TERRIFIC TIME FOR TWELVE MINUTES, ARE THE LIFE AND SOUL OF THE PUB, THEY ALL CLAP YOU ON THE BACK AND TELL YOU WHAT A GREAT CHAP YOU ARE AND THEN THE EARTH GETS UNEXPECTEDLY DEMOLISHED. YOU WAKE UP WITH A HANGOVER WHICH LASTS FOR ALL ETERNITY. YOU HAVE DIED.

Suppose, on the other hand, you decide to:

>EXIT THE VILLAGE PUB THEN GO NORTH

In that case you'll be off on the most mind-bogglingly hilarious adventure any earthling ever had.

You communicate – and the story responds – in full sentences. So at every turn, you have literally thousands of alternatives. If you decide it might be wise, for instance, to wrap a towel around your head, just say so:

>WRAP THE TOWEL AROUND MY HEAD

And the story responds:

THE RAVENOUS BUGBLATTER BEAST OF TRAAL IS COMPLETELY BEWILDERED. IT IS SO DIM IT THINKS IF YOU CAN'T SEE IT, IT CAN'T SEE YOU.

Simply staying alive from one zany situation to the next will require every proton of puzzle solving prowess your mere mortal mind can muster. So put down that beer and hitchhike down to your local software store today. Before they put that bypass in.

Comes complete with Peril Sensitive Sunglasses, a Microscopic Space Fleet, a DON'T PANIC Button, a package of Multipurpose Fluff and orders for the destruction of your home and planet.

Other interactive science fiction stories from Infocom.

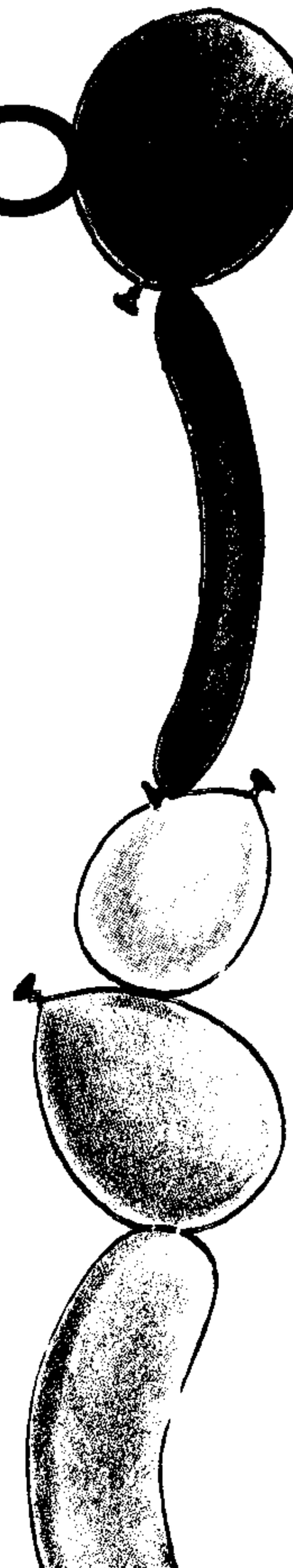
INFOCOM™

For more information call 1-800-262-6868. Or write to us at 125 CambridgePark Drive, Cambridge, MA 02140.

BALLOON CRAZY

Joseph Russ

Catch as many balloons as you can—but be careful not to fall off your skateboard. This whimsical game was originally written for Atari computers with at least 16K RAM. We've added versions for the Apple II series, Commodore 64, IBM PC (with color/graphics adapter and BASICA), IBM PCjr (with Cartridge BASIC), and TI-99/4A (with Extended BASIC). The 64, IBM, and Atari versions require a joystick. A joystick is optional with the TI version. The Atari and Apple versions can also be played with paddles.



"Balloon Crazy" is a game that children can enjoy, yet its higher levels are a challenge for adults. The goal is simple: You must zip back and forth across the screen on a skateboard while catching falling balloons on top of your head. Since some of the balloons fall very fast, that's not as easy as it sounds. After you've caught enough balloons (six in most versions), you can reach up to pop them, then catch some more. If you miss just one, you lose all the balloons currently in your possession.

Type in Balloon Crazy from the listing for your computer, then save a copy of the program before you try to run it. Every version of the game is similar, so be sure to read the general game rules before referring to the specific notes for your computer.

Oodles Of Balloons

Each game begins by displaying several rows of multicolored balloons at the top of the screen. You are the skateboarder at the bottom. When a balloon begins to fall, move directly under it and catch it on your head. The blue balloons fall slowly, which makes them easy to catch (but worth the fewest points). The green balloons fall faster, but swiftest of all are the red balloons. As soon as you snare a balloon, it joins the pile on top of your head.

Should you miss a balloon, you immediately fall off the skateboard. All the balloons on your head fall and pop. Points are scored only when you have caught the required number of balloons. You have three players to work with in each game: Falling off the skateboard costs you one player. Clearing all the balloons from a screen permits you to advance to the next level—where everything becomes more difficult. Bonuses are awarded at appropriate intervals, and you can earn an extra player by scoring 1,000 points.

Though the balloons become harder to catch at higher levels, you are never helpless to prevent them from hitting the ground. Should a balloon miss the top of the pile, you can bounce it back into the air by holding down the fire button (or the space bar in some versions) and running into it. The balloon will then float back into the air, and you may try to catch it again.

Atari Version

Balloon Crazy (Program 1) runs on any Atari computer with at least 16K RAM (or 24K for disk). Plug a joystick into port 1 before you run the game, and press START to begin. At the first level, you score 5 points for each green balloon, 10 for blue, and 15 for red. These values are multiplied at higher levels. A bonus player is awarded when you reach 1,000 points and at intervals thereafter. Move left and right with the joystick, and press the fire button when you want to hit a balloon. You must hit

awarded an extra player after completing level 5.

Apple Version

Apple II Balloon Crazy (Program 3) runs on Apple II-series computers with either DOS 3.3 or ProDOS. The listing must be entered using COMPUTE!'s "Apple MLX" machine language editor program found elsewhere in this issue. Be sure you understand the instructions for using Apple MLX before entering the data for Balloon Crazy. The MLX starting and ending addresses for the game are:

Starting address: 8000
Ending address: 8D97

After you've entered the game and saved a copy, start the game by entering:

```
BRUN "name"
```

where *name* is the filename you used when you saved Balloon Crazy.

You can play the game with a paddle on any Apple II computer: Move the paddle to control the player, and press the paddle button to bounce the balloon upward. Alternatively, keyboard controls can be used on the Apple IIc and Apple IIe: press the open-Apple key to move left, the closed-Apple key to move right, and the space bar to bounce.

Four balloons must be collected to score points. If you miss a balloon completely, all the balloons on your head drift off into space and disappear. There are nine game levels. Red balloons do not appear until the second level, but each higher level contains more red balloons. You may pause the game by pressing RETURN; resume play by pressing the space bar.

IBM PC/PCjr Version

IBM PC/PCjr Balloon Crazy (Program 4) requires a joystick and BASICA (if you have a PC) or Cartridge BASIC (PCjr). You may want to unlock the horizontal axis of the joystick. Before the game begins, you have an opportunity to adjust the joystick if needed: Press Y when prompted and follow the instructions on the screen. In this version, all balloons are red and are worth the same number of points.

The number of balloons you need to catch depends on how many rows of balloons are left on the screen: Only three are required at first, but this number increases each time you clear an entire row of balloons. When clearing the top row of balloons, you must catch seven balloons to score. There is no way to bounce a missed balloon back into play. After clearing an entire screen of balloons, you may advance to the next screen.

Your final score reflects the number of balloons caught (no bonus is awarded). You may adjust the difficulty of the game by changing the statement DF=10 in line 120. The variable DF controls how close you must be to a balloon to catch it. Changing DF to a higher value makes the game easier, and decreasing it makes the game more difficult.

TI-99/4A Version

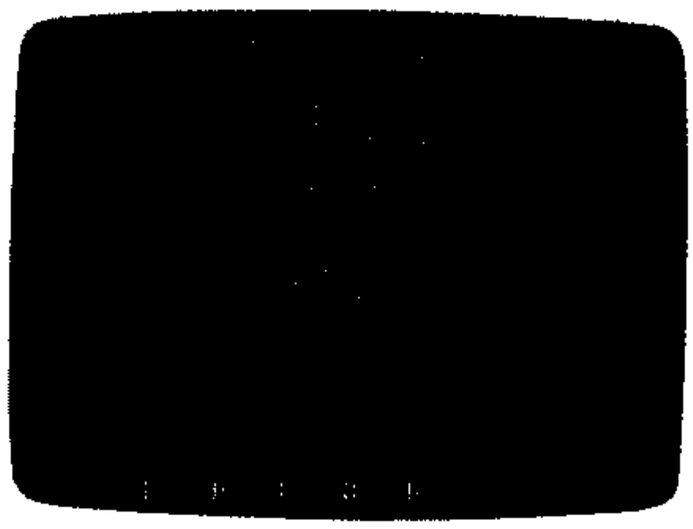
Balloon Crazy for the TI (Program 5) requires Extended BASIC and is played with either keyboard controls or a joystick. Press the S key to move left and the D key to move right. You cannot bounce a balloon back up after missing it. When you catch a balloon, it turns the same color as the player and immediately increases your score. At higher levels, the balloons fall faster and are worth more points. The game ends when you have lost all three players.

Program 1: Atari Balloon Crazy

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
JE 10  GOSUB 4500:GOSUB 5000:
       GOSUB 4000:GRAPHICS 17
       :POKE 756,CHS/256:POKE
        77,0:POKE 559,62:REM
       INITIALIZATION
CH 20  GOSUB 3500:GOSUB 3000
KK 30  GOSUB 2500
LP 40  FOR BY=BL TO 220 STEP
       SL:PM$(P1+BY,D+BY)=B$:
       GOSUB 500:GOSUB 1000:S
       OUND 0,BY,10,8:NEXT BY
       :SOUND 0,0,0,0
FP 50  BAL=BAL-1:GOSUB 1500:L
       F=LF-1:HIT=0:IF LF=0 T
       HEN 4100
IO 60  SOUND 0,0,0,0:PM$(P1+B
       Y,D+BY)=N$:HIT=0:POKE
       PC,1:GOSUB 1005:IF BOH
       =6 OR BAL<1 THEN GOSUB
        2000:BB=169:BOH=0
KF 70  IF BAL<1 THEN GOSUB 30
       10
AE 80  GOTO 30
PF 499 REM MOVEMENT
EC 500 S=STICK(0):PP=PP+((S=
       7)-(S=11)+(PP<65)-(PP
       >200))*3:POKE 53248,P
       P:RETURN
```

```
     BALL,PSET:PUT(50,63),MAN,
     PSET:FOR J=1 TO 100:NEXT:
     NEXT
NA 720 SKEW!=297/ABS(LX-TX)
MH 730 RETURN
PC 740 CLS:PUT(0,63),MAN:PUT (57
     ,0),BALL:FOR I=2 TO 50 ST
     EP 2:PUT (57,I-2),BALL:PU
     T (57,I),BALL:PUT (I-2,63
     ),MAN:PUT (I,63),MAN,PSET
     :NEXT:RETURN
HA 750 DATA &H2C,&H17,&H0,&H5,&H
     0,&H0,&H4015,&H0
FD 760 DATA &H0,&H5055,&H0,&H0,&
     H30CF,&H0,&H300,&HCC3
JJ 770 DATA &H0,&H300,&HACAA,&H0
     ,&H0,&HA0A0,&H0,&H0
BF 780 DATA &H802A,&H0,&H0,&HF,&
     H0,&HE00,&HEEEE,&HC0
EA 790 DATA &HFB00,&HBBBB,&HB0,&
     HC003,&HE0EE,&H3C,&HF,&HB
     03B
DM 800 DATA &HF,&HFF,&HC02E,&HF0
     0F,&H0,&H4015,&H0,&H0
GJ 810 DATA &HA0AA,&H0,&H0,&HA0A
     A,&H0,&H200,&HA8A0,&H0
FJ 820 DATA &H200,&HA8A0,&H0,&HA
     00,&H2A80,&H0,&HA37,&H2A8
     0
AB 830 DATA &HC00D,&HDADD,&H3AC0
     ,&H7077,&H7737,&H1D40,&HC
     0DD,&HA800
GP 840 DATA &H2C,&H17,&H800,&H5,
     &H0,&HA000,&H4015,&H0
IG 850 DATA &H8003,&H5055,&H0,&H
     C003,&HC3,&H0,&H30F,&H3CC
     F
KH 860 DATA &H0,&H30F,&HACAA,&H0
     ,&HC003,&HA0A0,&H0,&HF003
KI 870 DATA &H802A,&H0,&HFC00,&H
     F,&H0,&H2E00,&HEEEE,&HC0
CN 880 DATA &HB00,&HBBBB,&HBC,&H
     0,&HE0EE,&HFF,&H0,&H803B
EG 890 DATA &HF,&H0,&HC02E,&HC00
     3,&H0,&H4015,&HC003,&H0
GC 900 DATA &HA0AA,&HC000,&H0,&H
     A0AA,&H0,&H200,&HA8A0,&H0
FI 910 DATA &H200,&HA8A0,&H0,&HA
     00,&H2A80,&H0,&HA37,&H2A8
     0
AA 920 DATA &HC00D,&HDADD,&H3AC0
     ,&H7077,&H7737,&H1D40,&HC
     0DD,&HA800
PM 930 DATA &H3B,&H16,&H0,&H0,&H
     0,&H0,&H0,&H0
EG 940 DATA &H0,&H0,&H1400,&H0,&
     H0,&H0,&H55,&H0
OB 950 DATA &H0,&H55F1,&H4F,&H0,
     &HF303,&HCF3C,&HC0,&HF00
HN 960 DATA &HF30C,&HF0F0,&H0,&H
     F3C,&HF0AA,&H3C,&H3C00,&H
     820E
HN 970 DATA &H3CB0,&H0,&H23F,&H8
     028,&HFC,&HF00,&HFFC0,&HF
     003
BL 980 DATA &H0,&HFB03,&HBFBB,&H
     0,&H0,&HEE3E,&HEC,&H0
QJ 990 DATA &H300,&H80BB,&H0,&H0
     ,&HEE00,&H0,&H7700,&H0
PB 1000 DATA &HBB,&HDD00,&HC01D,
     &H5500,&H300,&H774,&HA24
     0,&HBAAA
LG 1010 DATA &HD001,&HE201,&HAAA
     A,&H8BAA,&H40,&HAA7A,&HA
     AAA,&HAD
DG 1020 DATA &H1A00,&H820A,&HA4A
     0,&H0,&H4,&H0,&H10,&HA00
     2
MJ 1030 DATA &H14,&HD,&HA802,&H2
     A00,&H80BE,&HAFAA,&HAAA0
     ,&HA0AF
MM 1040 DATA &HAFAA,&HAAA0,&HA0A
     F,&HAE2A,&H2A80,&H80AA,&
     HAA0A,&H200
MH 1050 DATA &HA8,&HA000,&H0,&H4
     0,&H1,&H200,&HA8
OP 1060 DATA &H26,&H12,&H2020,&H
     20,&H0,&H2800,&H0,&H0
NP 1070 DATA &H2828,&H800,&H2028
     ,&HA8,&H2A00,&HA002,&H28
     ,&H8202
DM 1080 DATA &H2080,&H202,&HA80,
     &H2000,&H0,&H8,&HA000,&H
     A088
DB 1090 DATA &H8,&H8028,&H2880,&
     H0,&H88A2,&HAA0,&H200,&H
     808A
FG 1100 DATA &HA000,&H2800,&H0,&
     H2000,&H0,&H200,&HB000,&
     H0
NN 1110 DATA &H80A,&HA0,&H400,&H
     2000,&H0,&H10,&H0,&H0
JE 1120 DATA &HE,&HA,&H1,&HC00F,
     &HC00E,&H3,&HB03B,&HCCCE
BD 1130 DATA &H1,&H800A,&HA028,&
     H1450,&H30CF
```



"Balloon Crazy" for the TI-99/4A can be played with the keyboard or a joystick.

Program 5: TI-99/4A Balloon Crazy

Version by Patrick Parrish, Programming Supervisor

```
90 REM REQUIRES EXTENDED
   BASIC
100 GOTO 140
110 CALL DELSPRITE(#2)::
    CALL MOTION(#1,0,0,#3
    ,0,0):: RETURN
120 CALL KEY(0,K,ST):: IF
     ST=0 THEN CALL JOYST
    (1,H,V):: H=SGN(H)ELS
    E H=(K=83)-(K=68)
130 CALL MOTION(#1,0,60*H
    ):: RETURN
140 DIM DROP(2),KOLOR(2):
    : RANDOMIZE :: CALL M
    AGNIFY(4)
150 CALL CHAR(136,"030303
    0301030703B0B0B0702020
    20F0480C0808000080C0A0
    9080808080F020"):::
    REM SKATEBOARD MAN
160 FOR I=96 TO 112 STEP
    8 :: CALL CHAR(I,"003
    87C7C7C381000"):: NEX
    T I :: LEVEL,SC,SC2=0
     :: MEN=3 :: ROW=41 :
    : KHAR=100
170 CALL CLEAR :: CALL SC
    REEN(16):: A$=RPT$("s
    hp",9):: FOR I=1 TO 2
    4 STEP 23 :: DISPLAY
    AT(I,1):A$ :: NEXT I
180 DISPLAY AT(10,8):"B A
     L L O O N" :: DISPLA
    Y AT(13,9):"C R A Z Y
     !" :: A=3 :: B=5 ::
    C=7
190 FOR I=1 TO 50 :: CALL
     COLOR(9,A,1,10,B,1,1
    1,C,1):: TEMP=A :: A=
    B :: B=C :: C=TEMP ::
     IF I=30 THEN CALL SP
    RITE(#1,136,14,150,1,
    0,31)
200 NEXT I :: CALL DELSPR
    ITE(#1):: CALL CLEAR
    :: GOSUB 490
210 DROP(0)=15 :: DROP(1)
    =20 :: DROP(2)=25
220 CALL CLEAR :: LEVEL=L
    EVEL+1 :: BALL=24 ::
    GOSUB 570
230 DISPLAY AT(1,6):"LEVE
    L:";LEVEL :: DISPLAY
    AT(1,17):"SCORE:";SC
240 FOR R=3 TO 6 :: FOR C
    =4 TO 29 STEP 5 :: CA
    LL HCHAR(R,C,96+INT(R
    ND*3)*8):: NEXT C ::
    NEXT R
250 CALL HCHAR(24,1,122,3
    2):: CALL SPRITE(#1,1
    36,14,150,115,0,H)
260 BALL=BALL-1 :: IF BAL
    L<0 THEN 410
270 BR=6 :: BC=4+INT(RND*
    6)*5
280 GOSUB 120 :: CALL GCH
    AR(BR,BC,BT):: IF BT=
    32 THEN BR=BR-1 :: IF
     BR=2 THEN 270 ELSE 2
    80
290 POINT=(BT-96)/8 :: CA
    LL HCHAR(BR,BC,32)::
    CALL SPRITE(#2,KHAR,K
    OLOR(POINT),ROW-(6-BR
    )*8,8*(BC-2)-2,DROP(P
    OINT),0)
300 GOSUB 120 :: CALL COI
    NC(#1,#2,15,C):: IF C
     THEN 340
310 CALL POSITION(#2,BROW
    ,BCOL):: IF BROW<155
    THEN 300
320 CALL POSITION(#1,MROW
    ,MCOL):: IF (BCOL-MCO
    L<16)*(BCOL-MCOL>-8)T
    HEN C=1 :: GOTO 340
330 GOSUB 110 :: MEN=MEN-
    1 :: CALL DELSPRITE(#
    3):: GOSUB 560 :: IF
    MEN=0 THEN 430 ELSE 4
    00
340 GOSUB 110 :: SC2=SC2+
    (POINT+1)*LEVEL*5 ::
    SC=SC+(POINT+1)*LEVEL
    *5
350 IF SC2>=1000 THEN MEN
    =MEN+1+(MEN=3):: SC2=
    0 :: GOSUB 570
360 IF C=0 THEN 400
370 CALL POSITION(#1,MROW
    ,MCOL):: CALL SPRITE(
    #3,100,14,118,MCOL)
380 FOR I=1 TO 50 :: NEXT
     I :: CALL SPRITE(#1,
    140,14,MROW,MCOL)
390 CALL SPRITE(#3,124,14
400 CALL HCHAR(1,3+MEN,32
    ):: DISPLAY AT(1,12):
    LEVEL;:: DISPLAY AT(1
    ,23):SC;:: GOTO 260
410 FOR G=300 TO 1200 STE
```

```
    P 100 :: CALL SOUND(8
    0,G,1):: NEXT G :: FO
    R I=0 TO 2 :: DROP(I)
    =DROP(I)+2 :: NEXT I
420 CALL DELSPRITE(ALL)::
     GOTO 220
430 CALL SCREEN(11):: IF
    SC>HS THEN HS=SC
440 CALL DELSPRITE(#1)::
    CALL CLEAR :: DISPLAY
     AT(8,5):"YOUR SCORE:
     ";SC :: DISPLAY AT(1
    1,5):"HIGH SCORE: ";H
    S
450 DISPLAY AT(16,5):"PLA
    Y AGAIN (Y/N)? " :: A
    CCEPT AT(16,24)BEEP V
    ALIDATE("NYny")SIZE(1
    ):REP$
460 IF REP$="N" THEN STOP
470 CALL SCREEN(16):: MEN
    =3 :: LEVEL,SC,SC2=0
    :: GOTO 210
480 REM SET COLORS
490 CALL COLOR(9,5,1,10,3
    ,1,11,7,1,12,13,1,13,
    14,1)
500 FOR J=0 TO 2 :: READ
    KOLOR(J):: NEXT J
510 DATA 5,3,7
520 CALL CHAR(100,"000000
    0000000000030707070
    3010000000000000000
    0080C0C0C0800000")::
    REM BALLOON
530 CALL CHAR(124,"000000
    0000000000010004010400
    0010000000000000000
    0000400040000000")::
     REM BALLOON POPPING
540 CALL CHAR(140,"030303
    0301030700B0B0B0702020
    20F0490D0909010A0C080
    8080808080808F20"):: R
    EM MAN POPPING BALLOO
    N
550 CALL CHAR(128,"383C38
    3810387CBA",122,"4949
    494949494949"):: RETU
    RN
560 FOR F=0 TO 25 STEP 5
    :: CALL SOUND(-200,-5
    ,F):: NEXT F :: RETUR
    N
570 CALL HCHAR(1,3,128,ME
    N):: RETURN
```

MEMO Diary

Jim Butterfield, Associate Editor

Keep track of important dates, holidays, and personal events with this simple, easy to use BASIC program. It was originally written for Commodore computers (with at least 8K RAM and a tape or disk drive), and modifications are included for the Atari 400/800, XL, and XE (with at least 16K RAM for tape or 24K RAM for disk); Apple II series (disk only, DOS 3.3 or ProDOS); IBM PC and Enhanced Model PCjr (disk only); and TI-99/4A with Extended BASIC (disk or tape).

"Memo Diary" helps you record and recall birthdays, holidays, appointments, or any other event worth remembering. The program maintains a data file with as many as 100 events whose dates can range from tomorrow to one year in the future. You can record two different types of dates: temporary, one-time events such as appointments which have no importance once they have passed; and permanent, recurring events such as birthdays and anniversaries. By routinely running Memo Diary each time you use your computer, you'll no longer have to worry about forgetting to mail a birthday card to a relative or finding an anniversary gift for a spouse.

The program always shows the correct day of the week when you enter a date, and you need to enter the year only once—the very first time you run the program. After that (for the next 99 years, anyway) Memo Diary keeps track of the year for you. Each time you run the program, it automatically shows all due and overdue events on the screen or printer, and erases one-time events from the calendar after they're displayed.

You can enter temporary or recurring new events and erase existing events whenever you wish. You can also examine all events from the current date forward, or search the entire calendar for events matching a given starting pattern. Finally, Memo Diary saves your calendar either on disk or tape.

Typing The Program

We've listed Memo Diary in the form of one main program that contains common routines (Program 1), followed by line changes for each different computer. No matter which computer you're using, you'll need to type in Program 1 plus the modifications for your machine. However, before typing anything, cross out every line in Program 1 that has the *same* line number as a line in the listing for your specific computer. The idea is to eliminate duplicate lines from the main program; they're replaced

by lines from the version for your computer. For example, if you're using an Atari, you would cross out line 150 in Program 1, because there's already a line 150 in the Atari listing (Program 3).

After crossing out duplicate lines in Program 1, type in the listing for your computer. Once that's done, type in every line of the Program 1 that's *not* crossed out. Be sure to save a copy of the program and read the instructions before running it.

The first time you run Memo Diary is special. *Do not start the program by entering RUN.* For every version except Atari you should type RUN 100 and press RETURN (or ENTER on the TI and IBM). Atari users should type CLR: GOTO 100 and press RETURN. *If you don't do this, the program will not work correctly.* When you start the program at line 100, Memo Diary lets you enter the correct year without looking for a previous file of events. Thereafter, start the program with RUN in the usual way.

On the first run you'll probably want to enter fixed holidays such as New Year's Day as well as birthdays and anniversaries. These are permanent events that you won't need to enter year after year. A holiday like Thanksgiving should be entered as a one-time event since it falls on a different date each year.

When Memo Diary asks you to enter today's date, you can type in the name of the month (such as OCTOBER) or its number (such as 10). In either case, be careful to enter it correctly. Memo Diary lets you enter any day of the month from 1 to 31, so it won't mind if you specify the date as February 30. Mistakes like these may confuse the calendar file. For instance, if you use the program on July 4 and the next day mistakenly give the date as June 5, the computer thinks you've let almost a whole year go by. To warn you of this, Memo Diary displays HAPPY NEW YEAR. If you see this message when a new year hasn't arrived, stop the program and start over, entering the correct date.

A Memory Jogger

Except for the very first run, Memo Diary always begins by reporting all due and overdue events ("You just missed your anniversary"). Take careful note of these events, since they'll soon be erased from the calendar (if they're temporary events) or moved ahead to next year (if they're permanent). To help jog your memory, Memo Diary also lets you make a copy of the list of events on your printer.

After disposing of due and overdue events, Memo Diary displays five options: You can see future events, add a new event, cancel an event, search for an event, or quit the program. You'll ordinarily want to look ahead to see what's coming in the next week or two. To do this, choose Option 1 (see future events) and supply an appropriate future date when requested. If you enter the current date when looking at future events, Memo Diary assumes you mean the same date *next year* and gives you everything on file.

When you want to make a new entry, select Option 2 (add new event). First Memo Diary asks whether the new event is one-time or permanent. Then it lets you enter the date and details. Again, the current date is understood as one year from today (it's assumed you don't need to record an event that's happening the same day).

To cancel an event (Option 3), you must know its date. When an event is entered, you're shown every item scheduled for that date, each with its own code number. To cancel an event, type in its code number when prompted.

Option 4 (search for event) lets you search for an event based on the first few letters of the entry. You may find many events in the course of a search. For instance, if the calendar file contains the events CLUB MEETING, CLUB CONFERENCE, and CLUB ELECTION, searching for CLUB displays all three events. In this case you would *not* see the entry CANADIAN CLUB, since CLUB is spotted only if it's in the first word of the entry. Thus, if you plan to search for certain keywords (BIRTHDAY, CHURCH, SOFTBALL, or whatever) keep them at the front of each calendar entry.

After you've finished an option, Memo Diary always returns you to the main menu. Sooner or later you'll be ready to use Option 5 (quit). The program knows when it's time to update the calendar file. If you've erased past and overdue events, added or deleted items, Memo Diary will—with your permission—proceed to update the data file on disk or tape.

The Time Pivot

A program that handles dates can encounter some subtle paradoxes. Does August come before April, or after it? The correct answer is *both*. Memo Diary could resolve this difficulty by adding a year designation to every event, but that complicates the handling of permanent events, which don't belong to a specific year. This is not a trivial problem: If you schedule a new event for August, the program must decide whether to add the event to the calendar ahead of an existing April event, or after it. Without a year designation, how can anyone tell?

The problem is solved by using a *pivot* date, usually the same as the current date. If today is July 4, August does indeed come before April. On the other hand, if today is November 11, April comes before August. Since the calendar always looks one year into the future, everything is kept in order.

However, there's one case in which the pivot date can't be the current date. Each time the program begins, it must measure the time lapse since its last use. For example, say that you last used the program on August 20, 1985 and next use it on September 4, 1985. On the first run (August 20) Memo Diary uses August 20 as the pivot. That way an event dated September 1 is seen ahead of another item dated in October.

On the second run (September 4) the September 1 event is reported as past due and either erased from the calendar (if it's temporary) or moved ahead to September 1 of next year (if it's permanent). Once this is done, the pivot date moves forward to September 4, meaning that a September 1 event now belongs *after* an item dated in October. Don't worry if this sounds confusing: It works out more simply in practice than in theory.

The day of the week is worked out with a simple formula. If you haven't seen it before, here's a hint on how it works. The calendar is

modified to make March 1 the first day of the "adjusted year." This way, leap year with its extra February 29 date doesn't break up the sequence of days: The extra leap day just gets pasted onto the year's end. Though the math is a bit convoluted, you may find it interesting to trace the logic of this routine (it starts at line 2150).

Expanding The Calendar

Memo Diary can keep track of a maximum of 100 events. In practice it's wise to limit the number to 80 or 90 to leave room for permanent events that move automatically from the front to the back of the list. If you need more than 100 events, change the L$ value in the DIM statement. Line 150 contains the value L$(100). You can increase the 100 to whatever number you like, but don't get carried away. Since Memo Diary (except the Atari version) uses string arrays, a very large value may cause garbage collection delays. There's no particular limit to the number of events allowed for a particular date.

Program Notes

Let's take a look at the program's major features. Line 90 prepares Memo Diary to read a file. The variable F is a *Boolean* (logical) variable that's defined as *true* here, to let you read the calendar file on a normal run. When you enter at line 100 on the first run, F is *false* (like every other undefined variable) and no file is read.

DATA statements in lines 110–140 hold the names of the months of the year and days of the week; the names are read into the arrays M$ and W$. Line 150 dimensions the L$ array for 100 items. Lines 230–250 call for a reading of the calendar file if appropriate. This is done in the subroutine at line 3010. When Memo Diary reads this file, it detects and reports the last date the file was used. Line 260 asks for today's date; the subroutine at line 1670 asks for and accepts the date.

Now it's time to search for due and overdue events. Using the previous date as a pivot, the subroutine at line 1960 scans for all events up to today's date. The program reports these events, erases them, or moves them ahead as needed, and proceeds to the main menu. Line 680 begins a main activity loop: It prompts with the menu, asks for a choice, then goes to the appropriate subroutine. Line 850 lets you see future events. Since the pivot date is now today, the program scans to the requested future date to see how many events fall into the today-to-future-date range.

Line 940 lets you add a new event. After asking ANNUAL OR ONE-TIME? the program requests the event's date and then asks for details. After adding a year designation to the date of one-time events, the new event is inserted into the proper sequence. Line 1210 lets you cancel an event. Memo Diary asks for a date and then lists all events that match that date. At line 1350, the program asks which event to delete. Note that the number you supply must be in the correct range.

Line 1450 begins the search-for-an-event routine. After it receives a search string (P$), the program looks for a match. When it scans through the calendar, it must look in different places depending on whether the event is one-time or permanent. That's because one-time events carry a year designation, making their dates three characters longer.

A Horrible Mistake?

Line 1570 handles the quit option; the flag F9 registers activity. If you haven't changed any of the data, there's no need to update the calendar file. Before scratching the old file and writing the new one, the program asks whether you're ready. That way, if you made some horrible mistake, you can cancel the file update.

The main loop ends at line 1580 and is followed by several subroutines. The routine starting at line 1590 writes a new calendar file when appropriate, and line 1670 begins the date input routine. The date is formed into a string (D8$) to allow for easy searches or entry. The subroutine at line 1930 reads the calendar file. The first item in the file is always the most recent date of use; the remaining data is events.

The subroutine at line 1960 scans all events to see which have dates between the pivot date (D9$) and a second date (D8$). There are three dates involved: event, pivot, and the second date, which makes the comparison a bit messy. Boolean variables keep everything in order. Eventually, the variable F0 indicates the date is in range, and the variable L0 indicates when the last event is found within the date range.

The routine starting at line 4020 displays the information, on the printer if desired. (TI users should change line 4070 to match their printer configuration.) The date is given complete with the day of the week, and events falling on the same day are grouped together. The weekday calculation begins at line 2150. The weekday variable, W, ranges from 0 to 6, so 0 means Sunday. As written, this routine is good for years ranging from 85 (1985) to 84 (2084). If you want to plan more than 99 years in advance, you'll need to modify the routine.

Program 1: Memo Diary Main Program

Please refer to instructions in the article before entering this listing.

```
90 F=(1=1)
100 GOSUB 2250
110 DATA JAN,FEB,MAR,APR,MAY,J
    UN
120 DATA JUL,AUG,SEP,OCT,NOV,D
    EC
130 DATA SUNDAY,MONDAY,TUESDAY
    ,WEDNESDAY
140 DATA THURSDAY,FRIDAY,SATUR
    DAY
150 DIM M$(12),W$(6),L$(100)
160 FOR J=1 TO 12
170 READ M$(J)
180 NEXT J
190 FOR J=0 TO 6
200 READ W$(J)
210 NEXT J
220 PRINT "EVENT CALENDAR"
230 IF F=0 THEN 260
240 C=1
250 GOSUB 3010
260 PRINT "TODAY'S DATE:"
270 Y8=Y9
280 GOSUB 1670
290 M8=M
300 D8=D
310 IF M8>=M9 THEN 330
320 Y8=Y9+1
330 IF M8<>M9 OR D8>=D9 THEN 3
    50
340 Y8=Y9+1
350 IF Y8<=Y9 THEN 370
360 PRINT "HAPPY NEW YEAR"
370 IF F THEN 400
380 PRINT "YEAR";
390 INPUT Y8
400 D9$=RIGHT$(STR$(100+M9),2)
    +"/"
410 D9$=D9$+RIGHT$(STR$(100+D9
    ),2)
```

```
420 IF F THEN 440
430 D9$=D8$
440 F=(1=1)
450 GOSUB 1960
460 PRINT "PAST EVENTS: ";
470 IF L0>=0 THEN 500
480 PRINT "NONE"
490 GOTO 650
500 PRINT L0+1
510 GOSUB 4010
520 F9=-1
530 FOR J=0 TO L0
540 IF MID$(L$(J),6,1)="/" THE
    N 570
550 L$(L9)=L$(J)
560 L9=L9+1
570 NEXT J
580 L8=L0+1
590 FOR J=L8 TO L9-1
600 L$(J-L8)=L$(J)
610 NEXT J
620 L9=L9-L8
630 L8=0
640 L=L9
650 F=0
660 F9=0
670 D9$=D8$
680 L=L9-L8
690 IF L<>0 THEN 710
700 PRINT "NO FUTURE EVENTS"
710 IF L=0 THEN 730
720 PRINT L;" FUTURE EVENTS"
730 PRINT
740 PRINT "1. SEE FUTURE EVENT
    S"
750 PRINT "2. ADD NEW EVENT"
760 PRINT "3. CANCEL EVENT"
770 PRINT "4. SEARCH FOR EVENT
    "
780 PRINT "5. QUIT"
790 PRINT
800 PRINT "...YOUR CHOICE (1-5
    )";
810 INPUT A
820 PRINT
830 ON A GOTO 850,940,1210,145
    0,1570
840 GOTO 730
850 PRINT "AHEAD TO DATE:"
855 FL=1
860 GOSUB 1670
865 FL=0
870 GOSUB 1960
875 IF D8$=D9$ THEN L0=L9-1
880 IF L0<>-1 THEN 910
890 PRINT "NO EVENTS"
900 GOTO 920
910 GOSUB 4010
920 PRINT L9-L0-1;" OTHER FUTU
    RE EVENTS"
930 GOTO 730
940 PRINT "ANNUAL OR ONE-TIME
    {SPACE}(A/O)";
950 INPUT P$
960 A=0
970 P$=LEFT$(P$,1)
980 IF P$="O" THEN 1010
990 A=1
1000 IF P$<>"A" THEN 730
1010 GOSUB 1670
1020 Y$="/"+RIGHT$(STR$(101+Y8
     ),2)
1050 IF A<>1 THEN 1070
1060 Y$=""
1070 GOSUB 1960
1080 IF L9-1<L0+1 THEN 1120
1090 FOR J=L9-1 TO L0+1 STEP -
     1
1100 L$(J+1)=L$(J)
1110 NEXT J
1120 PRINT "DETAIL";
1130 INPUT LL$
1140 D8$=D8$+Y$
1150 D8$=D8$+" "
1160 L$(L0+1)=D8$+LL$
1170 L9=L9+1
1180 L=L9
1190 F9=-1
1200 GOTO 680
1210 PRINT "CHANGE WHICH DATE:
     "
1220 GOSUB 1670
1230 L0=-1
1240 FOR J=L8 TO L9-1
1250 IF D8$<>LEFT$(L$(J),5) TH
     EN 1300
1260 L1=J
1270 IF L0<>-1 THEN 1290
1280 L0=J
1290 PRINT J;": ";L$(J)
1300 NEXT J
1310 IF L0<>-1 THEN 1340
1320 PRINT "NO EVENTS"
1330 GOTO 730
1340 PRINT
1350 PRINT " DELETE WHICH EVEN
     T ABOVE";
1360 INPUT A
1370 IF A<L0 OR A>L1 THEN 730
1380 FOR J=A TO L9-1
1390 L$(J)=L$(J+1)
1400 NEXT J
1410 L9=L9-1
1420 F9=-1
1430 PRINT " ... DELETED"
1440 GOTO 680
1450 PRINT "SEARCH FOR";
1460 INPUT P$
1470 P=LEN(P$)
1480 FOR J=0 TO L9-1
1490 A=7
1500 IF MID$(L$(J),6,1)<>"/" T
     HEN 1520
1510 A=10
1520 IF A+P-1>LEN(L$(J)) OR P$
     <>MID$(L$(J),A,P) THEN 15
     40
1530 PRINT L$(J)
1540 NEXT J
1550 PRINT "{4 SPACES}END OF S
     EARCH"
1560 GOTO 730
1570 IF F9<>0 THEN 1590
1580 END
1590 PRINT "READY TO WRITE NEW
      EVENTS FILE (Y/N)";
1600 INPUT P$
1610 IF LEFT$(P$,1)="Y" THEN 1
     630
1620 STOP
1630 D9$=D9$+"/"
1640 D9$=D9$+RIGHT$(STR$(Y8+10
     0),2)
1650 C=2
1660 GOTO 3010
1670 M=0
1680 PRINT "MONTH";
1690 INPUT MM$
1700 M=VAL(MM$)
1710 MM$=LEFT$(MM$+"XX",3)
1720 IF M=0 THEN 1760
1730 IF M<1 OR M>12 THEN 1670
1740 PRINT M$(M)
1750 GOTO 1810
1760 FOR J=1 TO 12
1770 IF MM$<>M$(J) THEN 1790
1780 M=J
1790 NEXT J
1800 IF M<1 OR M>12 THEN 1670
1810 PRINT "DAY";
1820 INPUT D
1830 IF D<1 OR D>31 THEN 1670
1840 D8$=RIGHT$(STR$(100+M),2)
     +"/"
1850 D8$=D8$+RIGHT$(STR$(100+D
     ),2)
1860 Y=Y8
1865 IFD8$=D9$ANDFL=1THEN1880
1870 IF D8$>=LEFT$(D9$,5) THEN
      1890
1880 Y=Y8+1
1890 GOSUB 2150
1900 IF LEN(LL$)<=0 THEN 1920
1910 PRINT "(";W$(W);")"
1920 RETURN
1930 C=1
1940 GOSUB 3010
1950 RETURN
1960 LL$=CHR$(255)
1970 L0=-1
1980 IF L<>0 THEN 2000
1990 RETURN
2000 V$=D8$+LL$
2010 WW$=D9$
2030 WW$=D9$+LL$
2040 F1=(WW$>V$)
2050 FOR J=L8 TO L9-1
2060 F2=(L$(J)>WW$)
2070 F3=(V$>L$(J))
2080 F0=F2 AND F3
2090 IF F1=0 THEN 2110
2100 F0=F2 OR F3
2110 IF F0=0 THEN 2130
2120 L0=J
2130 NEXT J
2140 RETURN
2150 IF Y>=85 THEN 2170
2160 Y=Y+100
2170 M1=M+1
2180 M2=INT(1/M1+.7)
2190 M3=Y-M2
2200 M4=M1+12*M2
2210 N=INT(M4*30.6001)+INT(M3*
     365.25)+D
2220 M6=INT(N/7)
2230 W=N-7*M6
2240 RETURN
2250 PRINT CHR$(147)
2260 RETURN
3000 REM INPUT/OUTPUT ROUTINE
4000 REM PRINT ROUTINE
```

Program 2: Modifications For Commodore

For instructions on entering this listing please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
255 IF E=0 THEN 260    :rem 164
256 F=0                :rem 80
1575 IFOTHENCLOSE15    :rem 187
3010 F$="EVENTS"       :rem 132
3020 PRINT"DISK OR CASSETTE (D
     /C)?"             :rem 4
3030 GETA$:IF((A$<>"C")AND(A$<
     >"D"))ORA$=""THEN3030
                       :rem 227
3040 IFA$="D"THEN3060  :rem 120
3050 D1=0:G$="":GOTO3070
                       :rem 13
3060 F$="@0:"+F$:D1=1  :rem 16
3070 IFC=2THEN3160     :rem 4
3080 IFD1=1THENG$=",S,R"
                       :rem 85
3090 OPEN1,1+7*D1,8*D1,F$+G$:G
     OSUB3220:IFETHENCLOSE1:GO
     TO3150            :rem 93
3100 INPUT#1,LL$:D9$=LL$:IF LE
     N(LL$)<>8 THEN PRINT LL$;
     "?":GOTO 3140     :rem 60
3110 M=VAL(LEFT$(LL$,2)):D=VAL
     (MID$(LL$,4,2)):Y0=VAL(MI
     D$(LL$,7,2))      :rem 245
3120 M9=M:D9=D:Y9=Y0:L=0:PRINT
      "LAST ACCESS: ";LL$
                       :rem 181
3130 INPUT#1,L$(L):L=L+1:IF ST
```

```
      GOTO 3060
76 3070 L$(L) = "":L8 = 0:L9 = L
      : GOTO 3090
42 3080 PRINT DD$;"WRITE ";F$: P
      RINT D9$: FOR J = 0 TO L
      9 - 1: PRINT L$(J): NEXT
       J: PRINT "EOF"
6A 3090 PRINT DD$;"CLOSE ";F$: I
      F C = 2 THEN END
D5 3100 RETURN
F8 4010 PRINT :D$ = "": INPUT "W
      ANT EVENTS ON PRINTER (Y
      /N) ";P$: IF LEFT$ (P$,1
      ) < > "Y" THEN 4030
E4 4020 PRINT DD$;"PR#1": PRINT
      I$;"80N"
6D 4030 FOR J = L8 TO L0: IF D$
      = LEFT$ (L$(J),5) THEN 4
      060
8A 4040 D$ = LEFT$ (L$(J),5):M =
       VAL ( LEFT$ (D$,2)):D =
       VAL ( MID$ (D$,4,2)):Y
      = Y8: IF D$ < = D9$ THEN
       Y = Y8 + 1
0A 4050 GOSUB 2150: PRINT W$(W);
      " ";M$(M);" ";D
B9 4060 PRINT "   "; MID$ (L$(J)
      ,6): NEXT J
D6 4070 PRINT : IF LEFT$ (P$,1)
      = "Y" THEN PRINT DD$;"PR
      #0"
F4 4080 RETURN
```

Program 5: Modifications For IBM PC/PCjr

For instructions on entering this listing please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
KL 105 WIDTH 80:KEY OFF:DEF SEG=
       0:POKE 1047,PEEK(1047) OR
        64
ND 2250 CLS
FD 3010 ON ERROR GOTO 3100
MA 3020 F$="EVENTS":INPUT "ENTER
         DRIVE # (IE., A): ";FF$
       :F$=FF$+":"+F$
EE 3030 IF C=2 THEN 3080
OK 3040 OPEN F$ FOR INPUT AS #1:
       INPUT #1,LL$:D9$=LL$:IF
       LEN(LL$)<>8 THEN PRINT L
       L$;"?":GOTO 3070
GE 3050 M=VAL(LEFT$(LL$,2)):D=VA
       L(MID$(LL$,4,2)):Y0=VAL(
       MID$(LL$,7,2)):M9=M:D9=D
       :Y9=Y0:L=0:PRINT "LAST A
       CCESS: ";LL$
FN 3060 INPUT #1,L$(L):L=L+1:IF
       EOF(1)=0 THEN 3060
PA 3070 CLOSE #1:ON ERROR GOTO 0
       :L8=0:L9=L:RETURN
CG 3080 OPEN F$ FOR OUTPUT AS #1
       :PRINT #1,D9$:FOR J=0 TO
        L9-1:PRINT #1,L$(J)
NB 3090 NEXT J:CLOSE #1:ON ERROR
        GOTO 0:END
EC 3100 CLOSE #1:PRINT "DISK ERR
       OR #";ERR;"OCCURRED.":PR
       INT "TRY AGAIN."
HH 3110 PRINT:PRINT "HIT A KEY T
       O CONTINUE"
FK 3120 A$=INKEY$:IF A$="" THEN
       3120
JA 3130 RESUME 3020
CJ 4010 ON ERROR GOTO 4090
BA 4020 D$="":INPUT "WANT EVENTS
         ON PRINTER (Y/N)";P$
PO 4030 IF LEFT$(P$,1)="Y" THEN
       OPEN "LPT1:" FOR OUTPUT
       AS #1 ELSE OPEN "SCRN:"
       FOR OUTPUT AS #1
CH 4040 FOR J=L8 TO L0:IF D$=LEF
       T$(L$(J),5) THEN 4080
BP 4050 D$=LEFT$(L$(J),5):M=VAL(
       LEFT$(D$,2)):D=VAL(MID$(
       D$,4,2))
JJ 4060 Y=Y8:IF D$<=D9$ THEN Y=Y
       8+1
GL 4070 GOSUB 2150:PRINT #1,W$(W
       );" ";:PRINT #1,M$(M);D
PL 4080 PRINT #1,"   ";MID$(L$(J
       ),6):NEXT J:CLOSE #1:ON
       ERROR GOTO 0:RETURN
IC 4090 CLOSE #1:PRINT "PRINTER
       ERROR #";ERR;"OCCURRED."
       :PRINT "TRY AGAIN."
HF 4100 PRINT:PRINT "HIT A KEY T
       O CONTINUE"
EG 4110 A$=INKEY$:IF A$="" THEN
       4110
KL 4120 RESUME 4020
```

Program 6: Modifications For TI-99/4A

```
330 IF (M8<>M9)+(D8>=D9)THE
    N 350
400 TE$=STR$(100+M9)
405 D9$=SEG$(TE$,LEN(TE$)-1
    ,2)&"/"
410 TE$=STR$(100+D9)
415 D9$=D9$&SEG$(TE$,LEN(TE
    $)-1,2)
540 IF SEG$(L$(J),6,1)="/"
    THEN 570
875 IF D8$<>D9$ THEN 880
876 L0=L9-1
970 P$=SEG$(P$,1,1)
1020 TE$=STR$(101+Y8)
1025 Y$="/"&SEG$(TE$,LEN(TE
     $)-1,2)
1040 TE$=STR$(100+Y8)
1045 Y$="/"&SEG$(TE$,LEN(TE
     $)-1,2)
1140 D8$=D8$&Y$
1150 D8$=D8$&" "
1160 L$(L0+1)=D8$&LL$
1250 IF D8$<>SEG$(L$(J),1,5
     )THEN 1300
1370 IF (A<L0)+(A>L1)THEN 7
     30
1500 IF SEG$(L$(J),6,1)<>"/
     " THEN 1520
1520 IF (A+P-1>LEN(L$(J)))+
     (P$<>SEG$(L$(J),A,P))T
     HEN 1540
1610 IF SEG$(P$,1,1)="Y" TH
     EN 1630
1630 D9$=D9$&"/"
1640 TE$=STR$(Y8+100)
1645 D9$=D9$&SEG$(TE$,LEN(T
     E$)-1,2)
1710 MM$=SEG$(MM$&"XX",1,3)
1730 IF (M<1)+(M>12)THEN 16
     70
1800 IF (M<1)+(M>12)THEN 16
     70
1830 IF (D<1)+(D>31)THEN 16
     70
1840 TE$=STR$(100+M)
1845 D8$=SEG$(TE$,LEN(TE$)-
     1,2)&"/"
1850 TE$=STR$(100+D)
1855 D8$=D8$&SEG$(TE$,LEN(T
     E$)-1,2)
1865 IF (D8$=D9$)*(FL=1)THE
     N 1880
1870 IF D8$>=SEG$(D9$,1,5)T
     HEN 1890
1960 LL$=CHR$(127)
2000 V$=D8$&LL$
2030 WW$=D9$&LL$
2080 F0=F2*F3
2100 F0=F2+F3
2250 CALL CLEAR
3010 F$="EVENTS"
3020 PRINT "DISK OR CASSETT
     E (D/C)?"
3030 CALL KEY(0,K,S)
3040 IF S=0 THEN 3030
3050 A$=CHR$(K)
3060 IF (A$<>"C")*(A$<>"D")
     THEN 3030
3070 IF A$="D" THEN 3100
3080 D$="CS1"
3090 GOTO 3110
3100 D$="DSK1."&F$
3110 IF C=2 THEN 3320
3120 OPEN #1:D$,INTERNAL,IN
     PUT ,FIXED
3130 INPUT #1:LL$
3135 D9$=LL$
3140 IF LEN(LL$)=8 THEN 317
     0
3150 PRINT LL$;"?"
3160 GOTO 3280
3170 M=VAL(SEG$(LL$,1,2))
3180 D=VAL(SEG$(LL$,4,2))
3190 Y0=VAL(SEG$(LL$,7,2))
3200 M9=M
3210 D9=D
3220 Y9=Y0
3230 L=0
3240 PRINT "LAST ACCESS: ";
     LL$
3250 INPUT #1:L$(L)
3260 IF L$(L)="EOF" THEN 32
     75
3270 L=L+1
3272 GOTO 3250
3275 L$(L)=""
3277 L=L-1
3280 CLOSE #1
3290 L8=0
3300 L9=L
3310 RETURN
3320 OPEN #1:D$,INTERNAL,OU
     TPUT,FIXED
3330 PRINT #1:D9$
3340 FOR J=0 TO L9-1
3350 PRINT #1:L$(J)
3360 NEXT J
3365 PRINT #1:"EOF"
3370 CLOSE #1
3380 END
4010 D$=""
4020 DE=1
4030 INPUT "WANT EVENTS ON
     PRINTER (Y/N) ":P$
4040 IF SEG$(P$,1,1)<>"N" T
     HEN 4070
4050 DE=0
4060 GOTO 4080
4070 OPEN #1:"RS232/2.BA=96
     00.DA=8.PA=N"
4080 FOR J=L8 TO L0
4090 IF D$=SEG$(L$(J),1,5)T
     HEN 4190
4100 D$=SEG$(L$(J),1,5)
4110 M=VAL(SEG$(D$,1,2))
4120 D=VAL(SEG$(D$,4,2))
4130 Y=Y8
4140 IF D$>D9$ THEN 4160
4150 Y=Y8+1
4160 GOSUB 2150
4170 PRINT #DE:W$(W);" ";
4180 PRINT #DE:M$(M);D
4190 PRINT #DE:"{3 SPACES}"
     ;SEG$(L$(J),6,LEN(L$(J
     )))
4200 NEXT J
4210 IF DE=0 THEN 4230
4220 CLOSE #DE
4230 RETURN
```

Reviews

Wishbringer

James V. Trunzo

Requirements: Commodore 64; Apple II-series computer with at least 48K RAM; Atari 400/800, XL, or XE with at least 48K RAM; IBM PC with at least 48K RAM; Expanded Model PCjr; Amiga; Atari 520ST; Macintosh; Kaypro CP/M; or a TRS-80 Model III. All versions require a disk drive. The Commodore 64 version was reviewed.

The latest entry from Infocom, the software industry's most prolific producer of text adventures, is a novel mystery/adventure entitled *Wishbringer*. It's billed as an introductory-level adventure, but veteran gamers should not be put off by the label. When Infocom calls a game "introductory," it simply means you might need only 20 or 30 hours to solve the adventure instead of 60 or 70 hours.

Actually, *Wishbringer* offers several very challenging puzzles, starting at the very beginning of the game when you have to map your way over the mountain leading to the Majick Shoppe. What makes *Wishbringer* slightly easier than a more advanced Infocom game is that some of the mapping is done for you, the scope of the storyline is not as broad, and the puzzles are slightly less devious. However, this should not be construed to mean that the game is child's play—far from it.

As the accompanying storybook says, you're in the role of an ordinary postal clerk in an "ordinary little town, and you've been performing your ordinary mail clerk's duties in an altogether ordinary way. But there's something quite extraordinary about today's mail." From that point your adventure begins, and nothing is the same any more.

A Piece Of The Rock

The adventure is twofold: First, you must seek out and obtain a magic stone known as the Wishbringer. To keep track of your location in the game's imaginary world, you should compile a map as you go along, even though a general map is included. If you find the Wishbringer, your second job is to use the powers of the stone (which are awesome in some ways, yet limited in others) to save your town—a town that no longer resembles what it was at the start of the adventure. Now it's filled with trolls, vultures, and other evil creatures.

Wishbringer conforms to the usual Infocom style. That is, it employs no graphics, relying on detailed descriptions and the player's imagination to provide the "pictures." The sophisticated parser, an Infocom trademark, lets you type in compound sentences rather than just primitive verb-noun commands. Other features let you save games in progress and send text to a printer. And as always with an Infocom package, *Wishbringer* is attractively designed. It includes a beautifully illustrated storybook, "The Legend of Wishbringer," and even a plastic Wishbringer stone that glows in the dark.

Starting with a simple premise—one that may seem almost childish at first—*Wishbringer* quickly becomes an enjoyable, playable adventure for all but the most hardened veterans of adventure games.

Wishbringer
Infocom
125 Cambridge Park Drive
Cambridge, MA 02140
$39.95

Remember!

Karen McCullough

Requirements: Commodore 64; Apple II-series computer with at least 64K RAM; IBM PC with at least 128K RAM and color/graphics adapter; or an Expanded Model IBM PCjr. All versions also require a disk drive. Joystick and printer optional.

Remember! bills itself as a "powerful, yet simple tool designed to help students from junior high through college master difficult subjects and improve memory skills." The claim is not exaggerated. This program presents an effective study system that teaches how to organize and memorize facts. It also helps you practice and test yourself on those techniques.

Two fundamental design principles give *Remember!* its power: You enter the facts you want to memorize only once, and the program then presents them to you in a variety of ways; and the program helps you build associations with the facts you're learning.

You begin by using the Create or Edit Lesson section to enter the facts you want to memorize into question and answer blocks. Once the lesson is entered, you can add hints to help you memorize the information. These hints can be in any of three formats: pictorial, musical, or written. Only one hint is allowed per question, and all hints for a given lesson must be in the same format. Editing functions allow you to make changes in the questions, answers, or hints at any time.

Entering hints is not quite as simple as entering the questions and answers. Although the program is generally flexible, drawing pictures or entering musical notation is not as intuitive as typing in questions. Both take some practice to master. One irritating aspect of entering pictorial or musical hints is that they are not automatically saved when you choose the Get Next Word option. This is the only time you must tell the program to save something, and it's easy to forget. (*Remember!* will remind you, however.)

Foreign Language Characters

Once the facts are entered, you have the option of reviewing them or testing yourself in various ways. The Familiarization option displays both questions and answers for review and study. When you feel thoroughly familiar with the material, you can choose the Practice option. In this mode, *Remember!* displays either the question or answer (your option), and you supply the missing part. If you can't remember the answer, pressing RETURN or Enter displays a hint (if you supplied one), and pressing the key again calls up the correct response. Finally, you can evaluate your progress with the Test option, which is similar to Practice mode.

Programming the TI

C. Regena

Christmas Graphics

Try this special Christmas program. (It can only be typed in on a TI-99/4A console.)

We Three Kings

```
100 REM  WE THREE KINGS
110 CALL CLEAR
120 T=375
130 CALL SOUND(2*T,494,2,39
    2,6,165,8)
140 CALL CHAR(152,"0001010303FF7F1F")
150 CALL CHAR(153,"0707ØF1F1C3Ø3Ø4")
160 CALL SCREEN(2)
170 CALL CHAR(154,"8Ø8ØC0C0E0FFFEF8")
180 CALL CHAR(155,"E0E0F0781Ø8ØC0C02")
190 CALL CHAR(33,"0000000515DF7F78")
200 CALL SOUND(T,440,2,370,
    6,165,9)
210 CALL CHAR(34,"020666606067773F1")
220 CALL CHAR(35,"0000000040C0E0F")
230 CALL SOUND(2*T,392,2,33
    0,6,165,8)
240 CALL CHAR(36,"E0C0D2929
    3333373")
250 CALL CHAR(37,"0107010001010101")
260 CALL CHAR(38,"50F0C0103
    0808Ø8")
270 CALL CHAR(39,"0100010307Ø704")
280 CALL CHAR(40,"F00080889
    8C86")
290 PRINT TAB(10);"!"
300 CALL SOUND(T,330,2,196,
    6,165,9)
310 CALL CHAR(41,"7B7B7BFBF
    BFBFBFB")
320 CALL CHAR(42,"808080E0F
    CF8F8F")
330 CALL SOUND(T,370,2,311,
    6,123,9)
340 CALL CHAR(43,"0303070604Ø103Ø7")
350 CALL CHAR(44,"90000020F
    0F0F0F")
360 CALL SOUND(T,392,2,311,
    7,123,9)
370 CALL CHAR(45,"0307070703
    70F0F0F")
380 CALL CHAR(46,"80C0C0C0E
    0E0E0F")
390 CALL SOUND(T,370,2,311,
    6,123,8)
400 CALL CHAR(47,"0000008081
    8183839")
410 CALL CHAR(48,"F9F9F8F8F
    9FDFDFC")
420 CALL SOUND(2*T,330,2,19
    6,6,165,8)
430 PRINT TAB(10);CHR$(34)
440 PRINT TAB(8);"# $  %&"
450 CALL CHAR(49,"F6E6E0400
    89CFEFE")
460 CALL CHAR(50,"0707070000
    E0F01")
470 CALL CHAR(51,"F0F0F8FC7
    E7C3911")
480 CALL CHAR(52,"0000000000
    01C3F8F")
490 PRINT TAB(7);"'( )* +,"
500 CALL CHAR(53,"0000000000
    030F8FC")
510 CALL CHAR(54,"0000010307070718")
520 CALL CHAR(55,"1F9FCFC7E
    7F3F9FC")
530 CALL SOUND(2*T,494,2,39
    2,6,165,8)
540 CALL CHAR(56,"F0F0E6F0E
    0C3CF1F")
550 CALL CHAR(57,"39190949E
    1F1F9F9")
560 CALL CHAR(58,"FDFDFDFDF
    DFDFDFD")
570 CALL CHAR(59,"7C0CE0FCF
    FFFFFFF")
580 CALL CHAR(60,"0001030303C3C7C7")
590 CALL SOUND(T,440,2,370,
    6,165,9)
600 CALL CHAR(61,"E7F3F8FCF
    EFFFFFF")
610 CALL CHAR(62,"C0E6FE7E3
    F1F8FC7")
620 CALL SOUND(2*T,392,2,33
    0,6,165,8)
630 CALL CHAR(63,"070F00383
    F000F3F")
640 CALL CHAR(64,"FFFFFF3F8
    000FEFE")
650 CALL CHAR(65,"80C0C0D01
    02C4C1C")
660 PRINT TAB(7);"-./01 234
    5"
670 CALL CHAR(66,"0303070309
    1C1F3F")
680 CALL CHAR(67,"FCFEFFFFF
    F1FC0CE")
690 CALL SOUND(T,330,2,196,
    6,165,9)
700 CALL CHAR(68,"FCFEFFFFF
    F1FC0CE")
710 CALL CHAR(69,"0000010101091939")
720 CALL SOUND(T,370,2,311,
    6,123,8)
730 CALL CHAR(70,"7BF9F9F8F
    8FCFEFF")
740 CALL CHAR(71,"F8F0E0C30
    70F1F3F")
750 CALL SOUND(T,392,2,311,
    5,123,8)
760 CALL CHAR(72,"9F9F9F9F9
    F9F9F9F")
770 CALL CHAR(73,"F9F9F9F1F
    1F1F3F3")
780 CALL SOUND(T,370,2,311,
    6,123,9)
790 CALL CHAR(74,"FFFFFFFFF
    FFF")
800 CALL CHAR(75,"E7E7E7EFE
    F8707Ø723")
810 CALL SOUND(2*T,330,2,19
    6,6,165,8)
820 CALL CHAR(76,"FFFFFFFFF
    FFFFFF")
830 CALL CHAR(77,"E1F0F8FCF
    CFEFEFE")
840 CALL CHAR(78,"3F3F3F3F3
    F3F3F3F")
850 PRINT TAB(6);"6789:;<=>
    ?@ABCD"
860 CALL CHAR(79,"FCF8F0E0C
    0C1C1C3")
870 CALL CHAR(80,"1800808Ø8
    ")
880 CALL CHAR(81,"3F7F00007
    FFFFFFF")
890 CALL CHAR(82,"FF930040C
    0818101")
900 PRINT TAB(5);"EFGHI:JKL
    MNOPQRS"
910 CALL SOUND(2*T,392,4,33
    0,8,165,10)
920 CALL CHAR(83,"383878108
    08")
930 CALL CHAR(84,"79F9F9F9F
    9710101")
940 CALL CHAR(85,"FFFFFFFEF
    EFCF8F")
950 PRINT TAB(5);"TUVHW:LXL
    YVZ[\]"
960 CALL CHAR(86,"3F7F7F7FF
    FFFFFFF")
970 CALL CHAR(87,"F3F3F3F3F
    3F3F3F3")
980 CALL SOUND(T,392,5,330,
    9,165,11)
990 CALL CHAR(88,"636171717
    078787C")
1000 CALL CHAR(89,"F7F7F7FC
    F8F8F1E1")
1010 CALL SOUND(2*T,440,4,3
    70,8,147,10)
1020 CALL CHAR(90,"C2828688
    8080802")
1030 CALL CHAR(91,"0101010303070F1F")
1040 CALL CHAR(92,"FEFCFCF8
    F8F8F8F")
1050 PRINT TAB(4);"^_`LHWaL
    bcdVeHf"
1060 CALL CHAR(93,"0303030604Ø4ØC18E")
1070 CALL CHAR(94,"0707070707070703")
1080 CALL SOUND(T,440,5,370
    ,9,147,11)
1090 CALL CHAR(95,"F9F9F9F9
    F9F9F9F9")
1100 CALL CHAR(96,"F8F0F0F0
    E0E4E4C4")
1110 CALL SOUND(2*T,494,3,3
    92,7,196,9)
1120 CALL CHAR(97,"FCFCFCFC
    FCFCFCFC")
1130 CALL CHAR(98,"6000000030F3F3F3F")
1140 CALL CHAR(99,"3F3F1F00
    80E1FBFA")
1150 PRINT TAB(4);"g_hLHijL
    Nklmno"
1160 CALL CHAR(100,"C3870F3
    F7F7F7F75")
```

```
1170 CALL CHAR(101,"337373F
     3F3F3F3F3")
1180 CALL SOUND(T,494,4,392
     ,8,196,10)
1190 CALL CHAR(102,"F0F0E0E
     0C0808")
1200 CALL CHAR(103,"2323636
     3C3C38383")
1210 CALL SOUND(T,587,2,392
     ,6,247,8)
1220 CALL CHAR(104,"C4C68E0
     E0E0E0E9E")
1230 CALL CHAR(105,"E7E7E7E
     7E0E0E0E5")
1240 CALL SOUND(T,523,2,370
     ,6,220,8)
1250 CALL CHAR(106,"FCFCFCF
     C000000F4")
1260 CALL CHAR(107,"F2F2F2F
     2F2E60404")
1270 CALL SOUND(T,494,2,392
     ,6,196,8)
1280 CALL CHAR(108,"7F7FFCF
     CF0C1031F")
1290 CALL CHAR(109,"8F0F1F1
     F1F9DBC38")
1300 CALL SOUND(T,440,3,370
     ,7,220,9)
1310 CALL CHAR(110,"F3F3F3F
     3F3F0F0F")
1320 CALL CHAR(111,"9E9C988
     808")
1330 CALL SOUND(T,494,3,392
     ,7,220.9)
1340 CALL CHAR(112,"070F1F3
     F7F7F7E7C")
1350 CALL CHAR(113,"F8F1E3C
     78E18")
1360 CALL SOUND(T,440,3,370
     ,7,220,9)
1370 CALL CHAR(114,"9C99830
     70F1F3F7F")
1380 CALL CHAR(115,"1F1F1F0
     0000070?")
1390 CALL SOUND(2*T,392,3,3
     30,7,247,9)
1400 CALL CHAR(116,"E5E5E50
     000387C78")
1410 CALL CHAR(117,"F4F4F40
     0000C1C7C")
1420 CALL CHAR(118,"FFFFFF0
     00000E0E")
1430 PRINT TAB(4);"pqrUstuv
     wxyz{"
1440 CALL CHAR(119,"3C3C3C3
     C3E3E3E3E")
1450 CALL CHAR(120,"C07F3F0
     3")
1460 CALL SOUND(T,370,3,311
     ,7,123,9)
1470 CALL CHAR(121,"FFFEFEF
     0000010101")
1480 CALL CHAR(122,"3830707
     0F0E0E0E")
1490 CALL SOUND(3*T,330,3,1
     96,7,165,9)
1500 CALL CHAR(156,"FF")
1510 CALL CHAR(157,"0102040
     81020408")
1520 CALL CHAR(158,"8080808
     0808080808")
1530 PRINT TAB(4);"! } ~"
1540 PRINT TAB(4);"!"
1550 CALL HCHAR(2,27,152)
1560 CALL HCHAR(3,27,153)
1570 CALL HCHAR(2,28,154)
1580 CALL HCHAR(3,28,155)
1590 CALL CHAR(123,"F87C7C3
     C3E1F1F0E")
1600 CALL CHAR(124,"3030303
     0303030303")
1610 CALL CHAR(125,"FFFCFCF
     8F0F0E0E")
1620 CALL SOUND(2*T,370,2,2
     94,6,220,8)
1630 CALL CHAR(126,"707070F
     0F0F0F0F8")
1640 CALL CHAR(127,"F8F1F1E
     1E1E0E0E")
1650 CALL CHAR(128,"FCF8F0F
     0F0F0F0F")
1660 CALL HCHAR(22,11,127)
1670 CALL HCHAR(22,12,128)
1680 CALL CHAR(129,"C0C0C0C
     0C0C0C0C")
1690 CALL CHAR(130,"1F1F0F0
     F0707070?")
1700 CALL SOUND(T,440,2,262
     ,8)
1710 CALL CHAR(131,"0103030
     3030303?")
1720 CALL CHAR(132,"C0C0C08
     0808")
1730 CALL SOUND(2*T,392,0,2
     94,6,196,8)
1740 CALL COLOR(16,16,1)
1750 CALL CHAR(133,"0E0C1C1
     C383060E")
1760 CALL CHAR(134,"0000010
     1030303?")
1770 CALL CHAR(135,"E0C0C08
     08")
1780 CALL HCHAR(22,13,129)
1790 CALL CHAR(136,"78783C1
     C0C030606")
1800 CALL CHAR(137,"6060404
     0404")
1810 CALL SOUND(T,392,1,294
     ,7,196,9)
1820 CALL HCHAR(2,23,156,3)
1830 CALL VCHAR(4,28,158,5)
1840 CALL HCHAR(4,26,157)
1850 CALL HCHAR(5,25,157)
1860 CALL SOUND(2*T,392,0,2
     94,6,196,8)
1870 CALL CHAR(138,"0303030
     1010?")
1880 CALL CHAR(139,"8080808
     0808080C0C")
1890 CALL CHAR(140,"0606030
     C0C1C1C3C")
1900 CALL HCHAR(22,14,130)
1910 CALL HCHAR(22,16,131)
1920 CALL CHAR(141,"0103070
     F0F0E0E0C")
1930 CALL CHAR(142,"C0C08")
1940 CALL SOUND(T,294,1,247
     ,6,196,8)
1950 CALL CHAR(143,"3038303
     C0E")
1960 CALL CHAR(144,"0603030
     F0F0707?")
1970 CALL SOUND(2*T,392,1,2
     47,7,165,9)
1980 CALL CHAR(145,"0000000
     00080C0E")
1990 CALL CHAR(146,"000080C
     0E0E")
2000 CALL CHAR(147,"E0F07038")
2010 CALL HCHAR(22,17,132)
2020 CALL HCHAR(22,18,133)
2030 CALL CHAR(148,"C0C0C0C
     0F07B3C1C")
2040 CALL CHAR(149,"3C1C0E0
     707")
2050 CALL SOUND(T,330,1,262
     ,6,131,8)
2060 CALL CHAR(150,"1C1C180
     8")
2070 CALL HCHAR(23,7,134)
2080 CALL HCHAR(23,8,135)
2090 CALL HCHAR(23,10,136)
2100 CALL HCHAR(23,11,137)
2110 CALL SOUND(2*T,392,1,2
     94,6,165,8)
2120 CALL HCHAR(23,12,124)
2130 CALL HCHAR(23,13,129)
2140 CALL HCHAR(23,14,138)
2150 CALL HCHAR(23,15,139)
2160 CALL HCHAR(23,16,140)
2170 CALL HCHAR(23,17,141)
2180 CALL HCHAR(23,18,142)
2190 CALL HCHAR(24,6,143)
2200 CALL HCHAR(24,7,144)
2210 CALL HCHAR(24,8,145)
2220 CALL HCHAR(24,10,138)
2230 CALL HCHAR(24,11,146)
2240 CALL HCHAR(24,12,143)
2250 CALL HCHAR(24,13,147)
2260 CALL HCHAR(24,15,148)
2270 CALL HCHAR(24,16,149)
2280 CALL HCHAR(24,17,150)
2290 CALL SOUND(T/2,9999,30
     )
2300 CALL SOUND(2*T,392,2,2
     94,7,165,9)
2310 CALL SOUND(T,392,4,294
     ,9,165,11)
2320 CALL SOUND(2*T,392,2,2
     94,7,165,9)
2330 CALL SOUND(T,294,2,247
     ,6,196,8)
2340 CALL SOUND(2*T,392,2,2
     47,6,165,8)
2350 CALL SOUND(T,330,2,262
     ,6,131,8)
2360 CALL SOUND(2*T,392,2,2
     94.6,196,8)
2370 CALL SOUND(T,9999,30)
2380 CALL SOUND(2*T,392,3,2
     47,7,165,9)
2390 FOR C=1 TO 15
2400 CALL COLOR(C,16,1)
2410 NEXT C
2420 CALL SOUND(T,392,4,247
     ,8,165,10)
2430 CALL SOUND(2*T,440,2,3
     70,6,147,8)
2440 CALL SOUND(T,494,2,370
     ,7,147,9)
2450 CALL SOUND(2*T,523,1,3
     92,5,131,8)
2460 CALL SOUND(T,494,1,392
     ,5,196,7)
2470 CALL SOUND(2*T,440,1,3
     92,6,147,8)
2480 CALL SOUND(T,494,2,370
     ,6,147,9)
2490 CALL SOUND(2*T,392,2,2
     47,6,196,8)
2500 CALL SOUND(T,392,3,294
     ,7,196,9)
2510 CALL SOUND(2*T,392,2,2
     47,6,196,8)
2520 CALL SOUND(T,294,2,196
     ,7,123,8)
2530 CALL SOUND(2*T,392,2,3
     30,6,131,8)
2540 CALL SOUND(T,330,2,262
     ,6,131,9)
2550 CALL SOUND(3*T,392,2,2
     94,6,247,9)
2560 CALL COLOR(16,12,1)
2570 CALL COLOR(16,16,1)
2580 CALL KEY(0,K,S)
2590 IF S<1 THEN 2560
2600 CALL CLEAR
2610 PRINT "HAVE A HAPPY HO
     LIDAY SEASON!":::::
2620 END                    ©
```