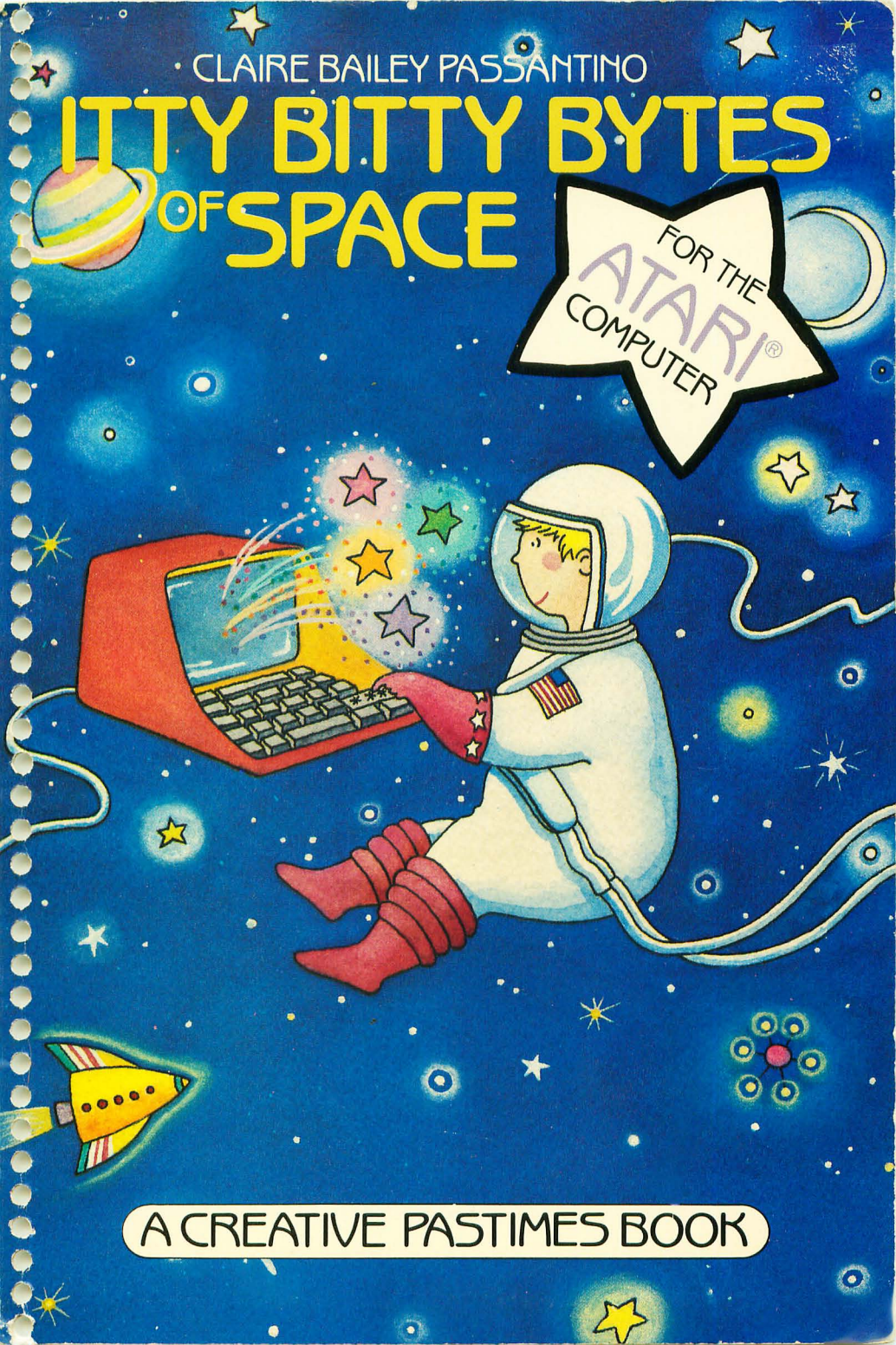


CLAIRE BAILEY PASSANTINO

ITTY BITTY BYTES OF SPACE

FOR THE
ATARI®
COMPUTER



A CREATIVE PASTIMES BOOK

This book belongs to

Itty Bitty Bytes of Space for the ATARI® Computer

Claire Bailey Passantino

Text Illustrations by Nancy Gurganus



**A Reston Computer Group Book
Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia**

A CREATIVE PASTIMES BOOK

Library of Congress Cataloging in Publication Data

Passantino, Claire Bailey.

Itty bitty bytes of space for the ATARI computer.

(A Creative pastimes book)

“A Reston Computer Group book.”

Summary: Twenty programs designed to reinforce beginning computer concepts and skills, including games, contests, races, songs, riddles, and other activities.

1. Atari 400 (Computer)—Programming—Juvenile literature. 2. Atari 800 (Computer)—Programming—Juvenile literature. 3. Basic (Computer program language)—Juvenile literature. 4. Computer programs—Juvenile literature. 5. Computer games—Juvenile literature. [1. Atari 400 (Computer)—Programming. 2. Atari 800 (Computer)—Programming. 3. Basic (Computer program language) 4. Computer programs. 5. Computer games. 6. Programming (Computers) I. Title. II. Series.

QA76.8.A8P37 1984

001.64'2

83-24580

ISBN 0-8359-3316-4

This book is published by Reston Publishing Company, Inc., which is not affiliated with Atari, Inc., and Atari is not responsible for the accuracy or any representation made herein. ATARI is a registered trademark of Atari, Inc. 400 and 800 are trademarks of Atari, Inc.

Cover illustration by Bethann Thornburgh

Cover design by Nancy Sutherland

© 1984 by Claire Bailey Passantino

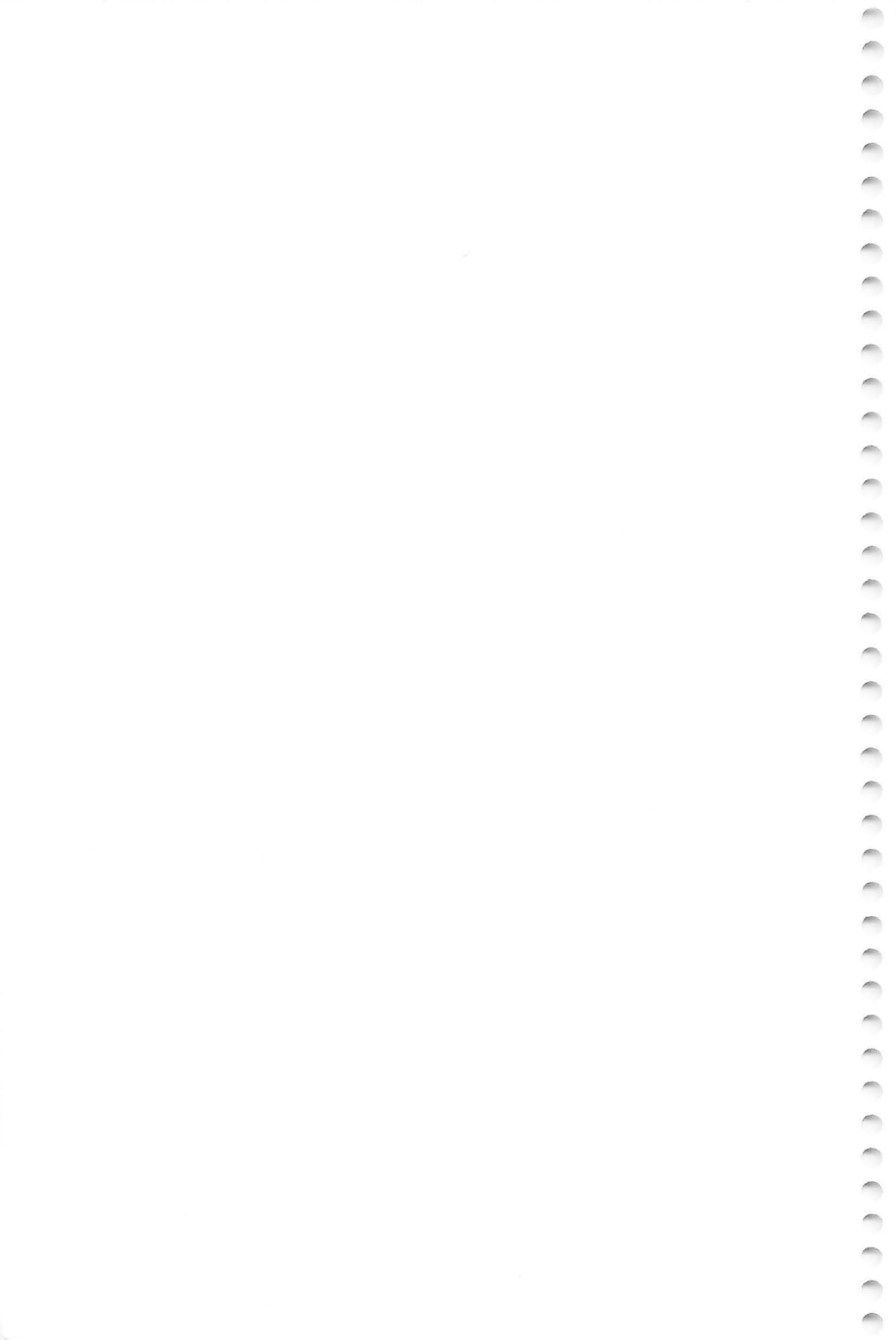
All rights reserved. No part of this book may be reproduced in any way, or by any means, without permission in writing from the author and the publisher.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America

Contents

Foreword, v
A Note to Parents and Teachers, vii
Passing the Test, 2–3
Gravity, 4–5
Test Your Spacecraft, 6–7
Take Off, 8–9
Count the Stars, 10–11
A Stellar Sight, 12–13
Music from Mother Earth, 14–15
Make Your Own Kind of Music, 16–17
What Is It?, 18–19
Who Is It?, 20–21
Hit or Miss?, 22–23
Nighty Night!, 24–25
Good Morning!, 26–27
Meteors, 28–29
How Much Fuel?, 30–31
Space Sketch, 32–33
Happy Landing!, 34–35
Final Report, 36–37
The Race for Space, 38–39
Break the Code, 40–41
Appendix, 43



Foreword

This is the first of the Itty Bitty Bytes Books for the ATARI® Computer.* The programs in this book were developed on the ATARI 400/800™ Computer.

Each book in the series is designed to bring you twenty fun programs and, at the same time, teach you something about BASIC. Typing in the programs is hard work. You must be careful to number the lines exactly, spell the words correctly, and put in all the right punctuation. But you will feel extra good when your program runs smoothly!

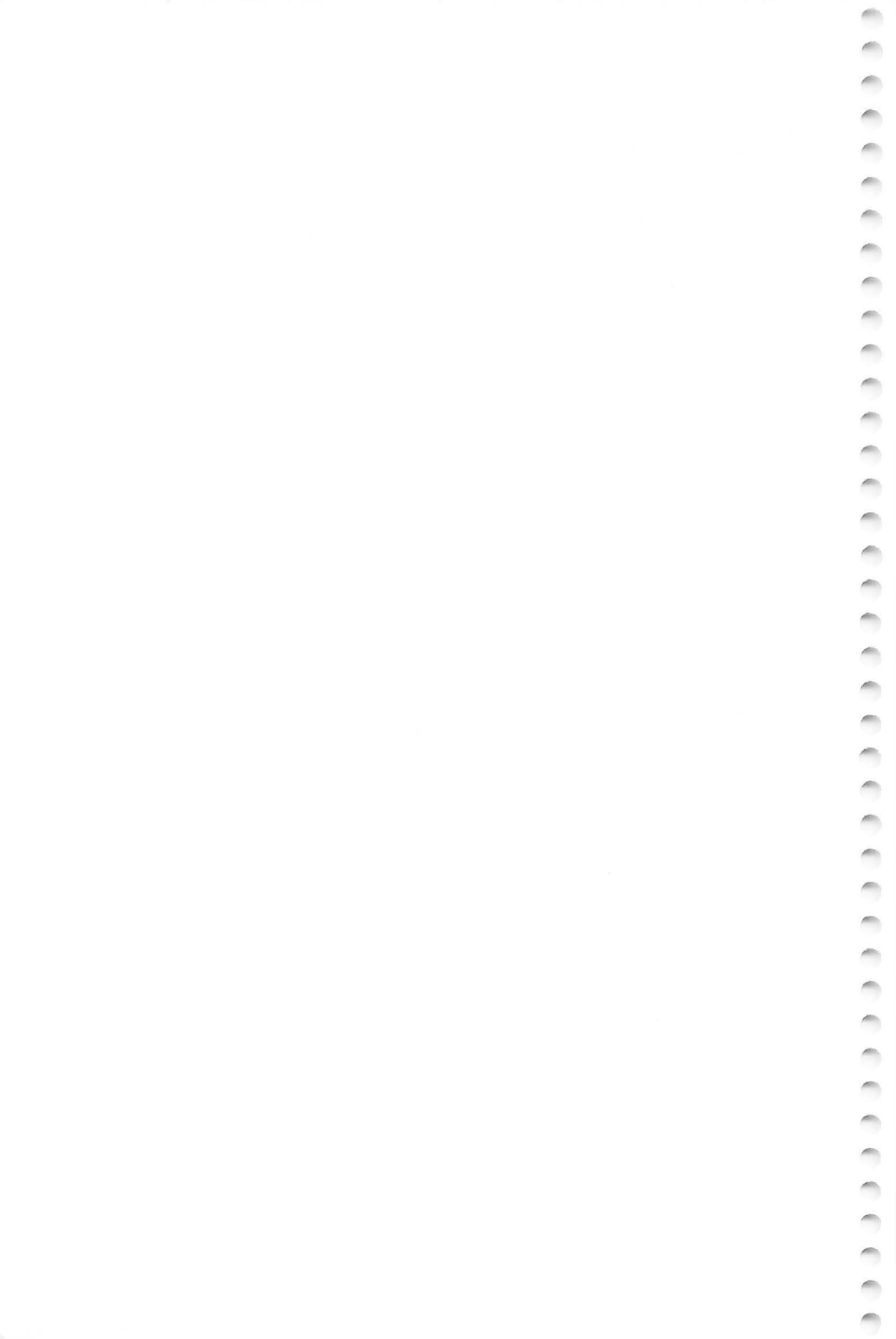
Always read the explanations next to each program. Sometimes there are additions or changes that you can make to improve the program. Once you see what the program is all about, feel free to experiment with it—and by all means combine programs that work well together. The more you can do to make a program reflect your own special personality, the more you will enjoy it. Let me know if you find some nifty changes. I love to get mail!

The Itty Bitty Bytes books have grown out of the teaching experiences I've had with my computer students. Special thanks go to all my "COMPU-KIDS"—and to my own kids, too—who keep presenting me with one good idea after another. They are my severest critics, but my most outstanding assets.

Happy computing!

Claire Bailey Passantino

*ATARI is a registered trademark of Atari, Inc. 400 and 800 are trademarks of Atari, Inc.



A Note to Parents and Teachers

You bought the computer. You read the manuals. You did the demo programs. You and/or your children may even have taken some computer classes. But now the computer is just sitting there. Everyone was so enthusiastic. *What happened?*

The novelty of a new computer will take you just so far. Beyond that point, a continuing interest in this incredible tool is directly related to its usefulness. "Not useful" equates to "not used." So the problem becomes, what can computers do that children would find useful? What kinds of things would encourage children to expend the energy needed to create their own computer programs?

Each Itty Bitty Bytes book in the Creative Pastimes series is packed with computer activities appealing to young programmers. Simple games, contests, races, pictures, designs, songs, riddles, charts, tests, and more—all are designed to be fun while reinforcing beginning computer concepts and skills.

Besides providing fun, there are fringe benefits to having children write their own programs. In working with children, I have found that computer programming encourages them to:

- Think creatively.
- Use logical thinking skills.
- Attend to details.
- Take small steps to achieve a goal.
- Personalize programs.
- Develop pride and self-esteem.
- Appreciate packaged software.

To help children enjoy doing their own programs, here are some suggested DOs and DON'Ts:

- DO encourage children to type in their own programs. With younger children, bargain: "You type this line and I'll type the next one." (Save the long lines for yourself!)
- DON'T criticize typing expertise. Speed and correct fingering are typing skills that are minimally related to computer programming.
- DO allow children to make mistakes.
- DO help them find the errors they've made. (This is called "debugging" the program.)
- DON'T worry when you get an error message. This means you've made a mistake. Check your program for "bugs." (The ATARI Computer will give you an "ERROR" when you go over a "READY" signal. Ignore this message. The computer is trying to "READ Y" and there is no "Y" to read.)

-
- DO encourage children to read and understand the program explanations.
 - DON'T, however, force the issue. Some people learn by reading. Others learn by doing. As skills are repeated over and over in different contexts, children may just "catch on."
 - DO be aware of some common pitfalls. Remember to:
 1. Use line numbers.
 2. Press RETURN after you type in a line.
 3. Give great attention to spelling and punctuation. Quotation marks, commas, semicolons, colons, and even spaces are often important.
 4. Save your program before you turn off the computer.
 - DO help children save their programs on tape or disk so they can use them again and again. If you have a printer, use it to make "hard" copies of each program. People like to see themselves in print.
 - DON'T enter commands in reversed or lower-case characters. (Reversed characters are produced by pressing "↵.") Press ↵ again to return to regular characters. Lower-case letters are produced by pressing the "CAPS/LOWR" key. To return to capital letters, press the "SHIFT" and the "CAPS/LOWR" keys simultaneously.
 - DO praise children for a job well done. And enjoy the programs that they've created.
 - DO modify and use programs that you yourself find useful.

It is my sincere hope that the Itty Bitty Bytes books will help you and your children establish a healthy working relationship with your computer. Take that computer off the shelf! And let me know how things are working out!

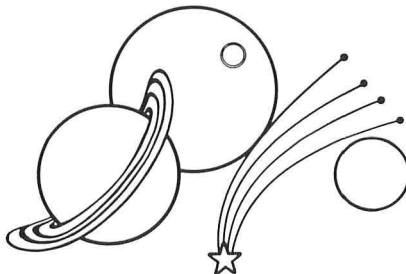
Itty Bitty Bytes of Space
for the ATARI® Computer

Main ideas: A test using simple data statements
Scoring the test

C = Number correct
T = Total number of questions
PL\$ = Name of planet
F\$ = Fact about planet
A\$ = Answer
SC = Score

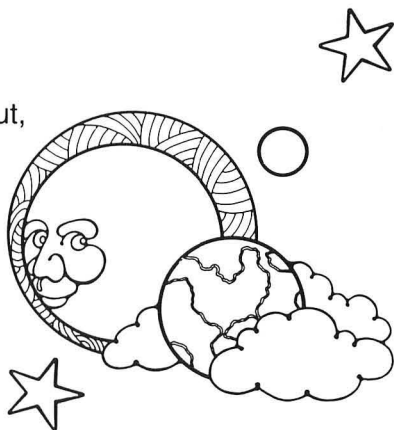
- 10 Sets the variables to 0.
- 20 Dimensions the string variables.
- 30 Pauses.
- 40 Clears the screen.
- 50 Reads the name of the planet and the fact about it (see lines 200 to 290).
- 60 Tests for the end of the data (see line 300).
- 70 Prints the fact.
- 80–90 A chance to guess the answer.
 - 95 The total number of questions goes up by 1.
- 100 If the answer is right, the computer adds 1 to the number correct and goes to line 30 for the next question.
- 110 If the answer is wrong, it prints the correct answer and goes to line 30 for the next question.
- 120 Computes the score.
- 130 If the score is less than 90, you fail.
- 140 If the score is 90 or better, you pass.
- 150 Prints the score.
- 200–290 Data.
 - 300 Flag data.

If you know more facts about planets, you may add them to your data list. Or use this program to make up a test about something else!



Passing the Test

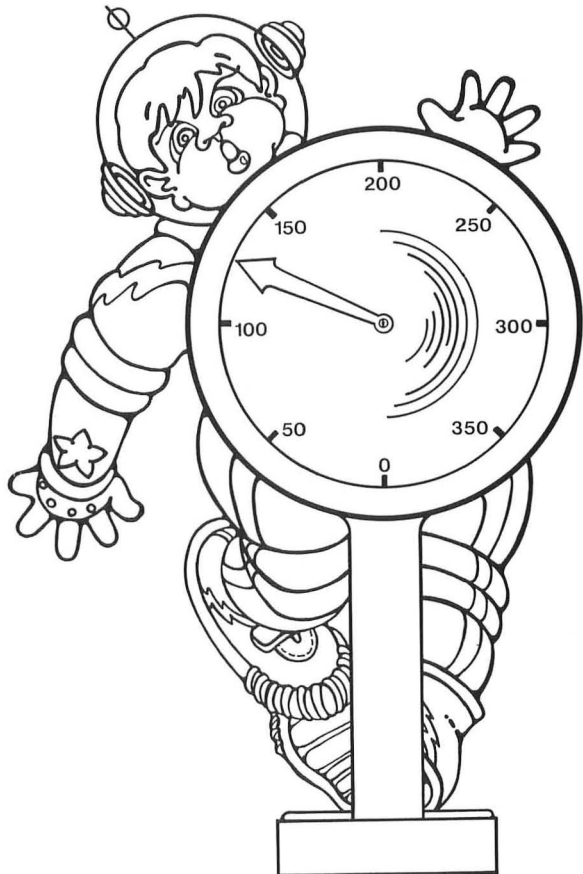
How well do you know your planets? To be an astronaut, you must pass this test. A score of 90 is passing. Good luck!



```
10 C=0:T=0
20 DIM PL$(10),F$(30),A$(10)
30 FOR P=1 TO 500:NEXT P
40 GR.0
50 READ PL$,F$
60 IF PL$="END" THEN GOTO 120
70 PRINT F$
80 PRINT "WHAT IS THE PLANET?"
90 INPUT A$
95 T=T+1
100 IF A$=PL$ THEN PRINT "RIGHT!":C=C+1:GOTO 30
110 IF A$<>PL$ THEN PRINT "WRONG, THE ANSWER
    WAS ";PL$:GOTO 30
120 SC=INT(C/T*100+.5)
130 IF SC<90 THEN PRINT "SORRY. YOU DID NOT PASS!"
140 IF SC>=90 THEN PRINT "YOU PASSED THE TEST!"
150 PRINT "YOUR SCORE WAS ";SC
200 DATA SATURN,THE PLANET WITH RINGS
210 DATA EARTH,OUR PLANET
220 DATA VENUS,THE MORNING AND EVENING 'STAR'
230 DATA MARS,THE RED PLANET
240 DATA JUPITER,THE LARGEST PLANET
250 DATA PLUTO,THE FARTHEST PLANET
260 DATA MERCURY,NEAREST TO THE SUN
270 DATA NEPTUNE,NAMED AFTER THE GOD OF THE SEA
280 DATA URANUS,7TH FROM THE SUN
290 DATA PLUTO,A FAMOUS DOG
300 DATA END,END
```

Main idea: Working with a simple conversion formula
Setting up a chart

- 10 Clears the screen.
- 20–30 A chance to input your Earth weight.
- 40 Clears the screen again.
- 50 Prints your weight on Earth.
- 60–90 Positions and prints the title of the chart and the headings.
- 100 Skips a line.
- 110 Prints the name of each planet; then computes and prints your weight on that planet. The ATARI has 4 print zones. The comma after PRINT moves the next item printed to the second print zone.



Gravity

Surface gravity varies on the planets. As part of your training, you must experience various gravitational pulls.

```
10 GR.0
20 PRINT "HOW MUCH DO YOU WEIGH ON EARTH?"
30 INPUT W
40 GR.0
50 PRINT "WEIGHT ON EARTH: ";W
60 POS.8,5:PRINT "WEIGHT ON OTHER PLANETS"
70 POS.8,6:PRINT "-----"
80 PRINT,"PLANET","WEIGHT"
90 PRINT,"-----","-----"
100 PRINT
110 PRINT,"MERCURY",.37*W
120 PRINT,"VENUS",.876*W
130 PRINT,"MARS",.381*W
140 PRINT,"JUPITER",2.637*W
150 PRINT,"SATURN",1.151*W
160 PRINT,"URANUS",.79*W
170 PRINT,"NEPTUNE",1.12*W
180 PRINT,"PLUTO",.025*W
```

How much would you weigh on Pluto?
on Venus? on Jupiter? Find out!

Main ideas: Printing on the Graphics 2 screen
Infinite loop

- 10 Clears the screen.
- 20 Dimensions the string variables (N\$ = Name, A\$ = Answer).
- 30–40 A chance to enter your name! When you enter your name, try using lower-case letters for light green, reversed letters for blue, and reversed lower-case letters for red.
- 50 Graphics 2 screen.
- 60 Positions the message.
- 70–90 Prints "HELLO" on the screen.
- 100–110 A chance to answer the question in the text window.
- 120 Prints on the screen.
- 130 Prints in the text window.
- 140 Goes to line 130 and puts this program into an infinite loop. You must press BREAK or SYSTEM RESET to stop the program.

Note: You may change the background color by adding this line:

55 SETCOLOR 4, —, —

a color from the Setcolor Chart (0 to 15)

a shade of color, dark (0) to light (15)

Example: 55 SETCOLOR 4, 5, 6

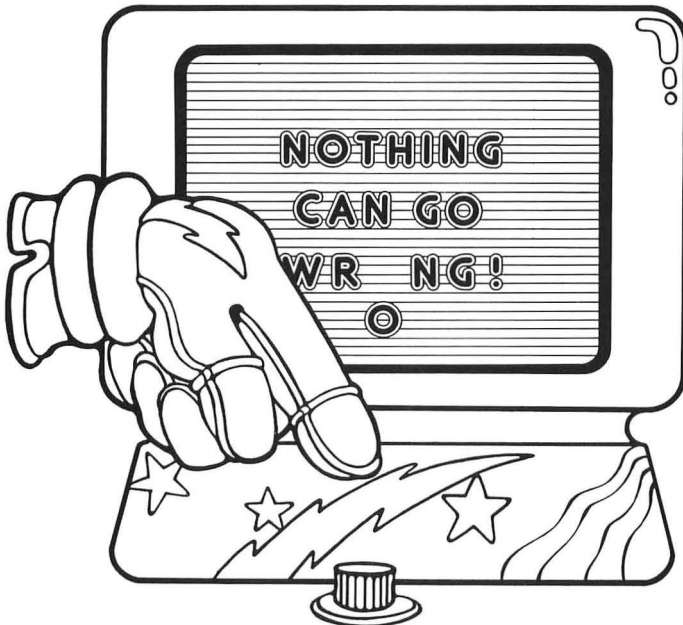
Setcolor Chart

- 0 black
- 1 rust
- 2 red-orange
- 3 dark orange
- 4 red
- 5 dark lavender
- 6 cobalt blue
- 7 ultramarine blue
- 8 medium blue
- 9 dark blue
- 10 blue-gray
- 11 olive green
- 12 medium green
- 13 dark green
- 14 orange-green
- 15 orange

Test Your Spacecraft

You are about to take off into outer space. Before embarking on your journey, run this program to test your spacecraft. Don't worry! Nothing can go wrong!

```
10 GR. 0
20 DIM N$(20), A$(3)
30 PRINT "WHAT IS YOUR NAME?"
40 INPUT N$
50 GR. 2
60 POS. 0, 3
70 PRINT #6; "HELLO"
80 PRINT #6; "CAPTAIN"
90 PRINT #6; N$
100 PRINT "ARE YOU READY TO GO?(Y,N)"
110 INPUT A$
120 PRINT #6; "NOTHING CAN GO WRONG"
130 PRINT "WRONG",
140 GOTO 130
```



Main ideas: Using RND for color and sound
Fading loudness
Dummy line

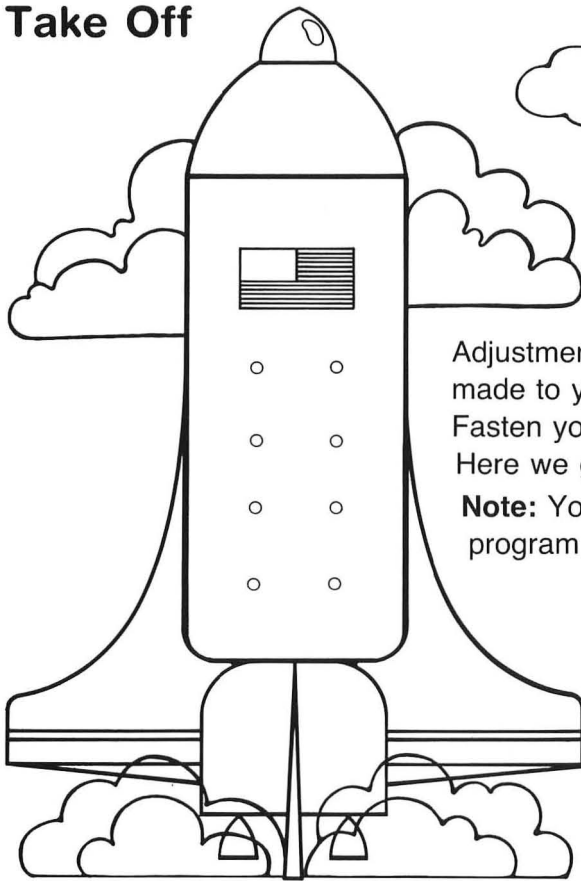
Blastoff:

- 10 Graphics 2 screen—add +16 to remove the text window.
- 20–30 Positions and prints a message on the screen.
- 40–90 A FOR . . . NEXT loop which repeats 50 times.
 - 50 Picks a random color (C), 0 to 15.
 - 60 Picks a random shade (SH), 0 to 15.
 - 70 Changes the background color.
 - 80 Makes a random sound. (Since C is 0 to 15 and sounds go 0 to 255, we can multiply C by 15 to get a wider range of sounds.)
- 100 A dummy line keeps the program from ending. (What happens if you omit this line?)

Liftoff:

- 100–120 “Counts” the sounds from 255 to 1 and plays them.
- 130–160 The loudness of the last sound goes from 15 (loudest) to 0 (softest).
 - 150 Pauses. If omitted, the sound fades too fast. The pause slows the program down so the sound fades gradually.
- 200 A new dummy line keeps the program from ending.

Take Off



Adjustments have been made to your spacecraft. Fasten your seat belt. Here we go!

Note: You may enter this program in two parts!

Blastoff

```
10 GR.2+16
20 POS.4,5
30 PRINT #6;"BLAST OFF!!!"
40 FOR B=1 TO 50
50 C=INT(16*RND(1))
60 SH=INT(16*RND(1))
70 SE.4,C,SH
80 SO.0,15*C,8,8
90 NEXT B
100 GOTO 100
```

Liftoff

```
100 FOR S=255 TO 1 STEP -1
110 SO.0,S,8,8
120 NEXT S
130 FOR L=15 TO 0 STEP -1
140 SO.0,S,8,L
150 FOR P=1 TO 200:NEXT P
160 NEXT L
200 GOTO 200
```

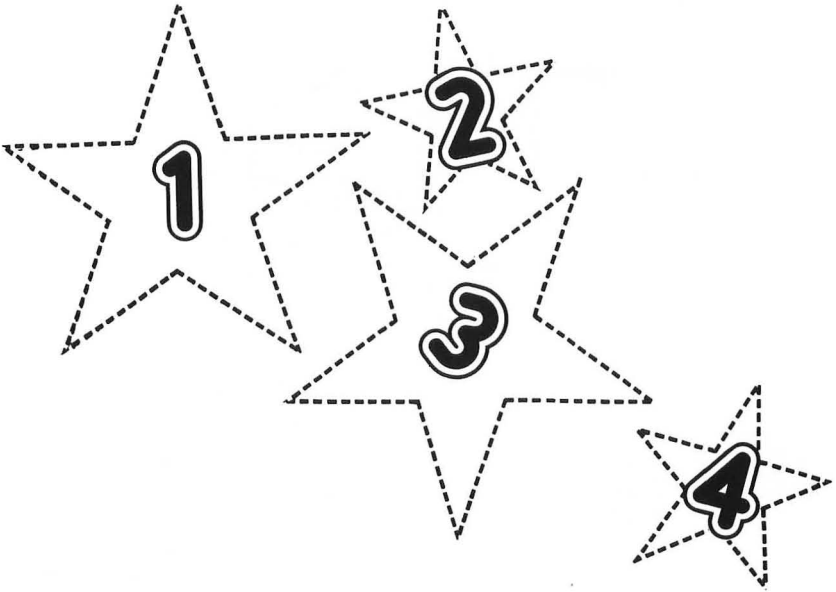
Main idea: Counting by incrementing

- 10 Clears the screen.
- 20–60 Positions and prints a message.
- 70 Pauses.
- 80 Sets the counter to 1.
- 90 Prints the number and a star.
- 100 Makes the counter go up 1 (adds 1 to the counter).
- 110 Goes to line 90 to make an infinite loop.

To count by twos:

```
80 N = 2
90 PRINT N; "*"
100 N = N + 2
```

Can you fix these lines so that the computer will be able to count by threes?



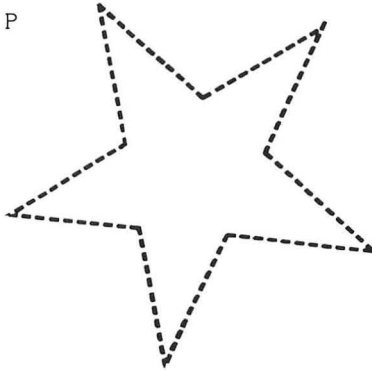
Count the Stars

It's amazing how many stars you see!
Start to count them.

Press BREAK when you get tired of counting.
Type CONT to continue counting!



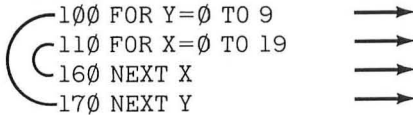
```
10 GR.0
20 POS.5,8
30 PRINT "TWINKLE TWINKLE LITTLE STAR"
40 POS.5,10
50 PRINT "HOW I WONDER HOW MANY THERE ARE!"
60 PRINT
70 FOR P=1 TO 500:NEXT P
80 LET N=1
90 PRINT N;" * ";
100 N=N+1
110 GOTO 90
```



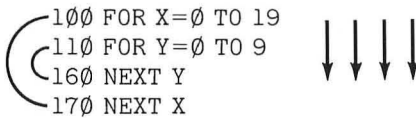
Main ideas: Random placement on the screen
Moving horizontally and vertically

- 5 Graphics 2 screen.
- 100 Start of the Y loop. The Ys go from 0 to 9.
- 110 The cursor will move across the Xs from 0 to 19.
- 120 At each point, the computer picks a number from 0 to 7. (Your chance of getting a star is 1 out of 8.)
- 130–150 If it selected 0, then at the current position it prints a big star. If it didn't select 0, it skips to line 160.
- 160 Continues to move along the X line to 19.
- 170 When all the Xs are finished, the cursor moves down to the next Y. In effect, this program moves across one row at a time, then down to the next row.

In this program, we have the X loop nested inside the Y loop, like this:



If you put the Y loop inside the X loop, the computer will move down the columns instead of across the rows:

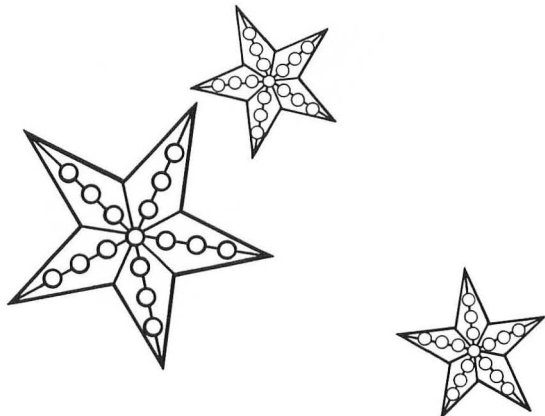


To try this exercise in Graphics 1, change line 5:

```
5 GR. 1
```

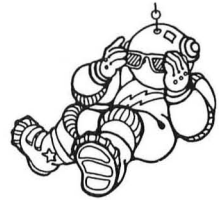
and change the Y line:

```
FOR Y=0 TO 19
```

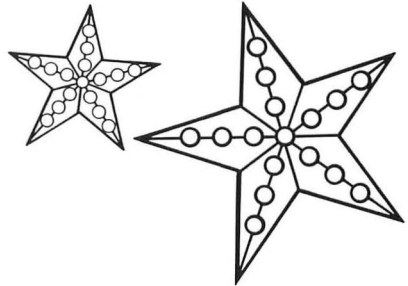


A Stellar Sight

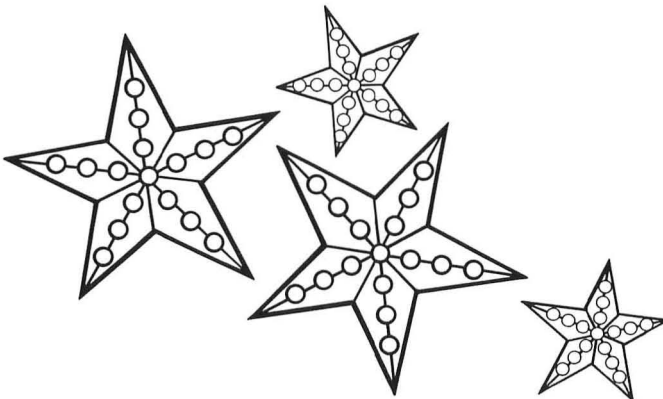
The stars are truly beautiful.
See for yourself!



```
5 GR.2
100 FOR Y=0 TO 9
110 FOR X=0 TO 19
120 R=INT(8*RND(1))
130 IF R<>0 THEN 160
140 POS.X,Y
150 PRINT #6;"*"
160 NEXT X
170 NEXT Y
```



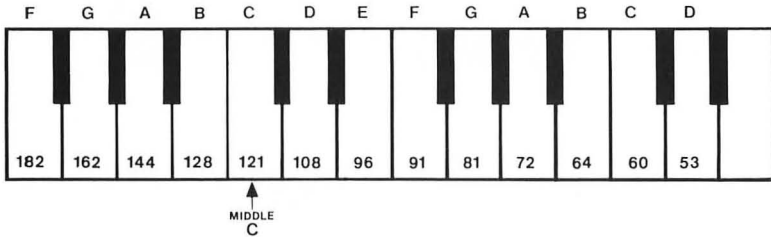
For fun, add the next program to this one.
They work well together!



Main idea: How to write a song

In this program, N = Note

T = Time to hold the note (number of beats)



The numeric equivalents of some of the notes are shown in the piano diagram. To play this piece on the piano, you would play:

Note N for T beats

C(121) for 1 beat

C(121) for 1 beat

G(81) for 1 beat

G(81) for 1 beat

A(72) for 1 beat

etc.

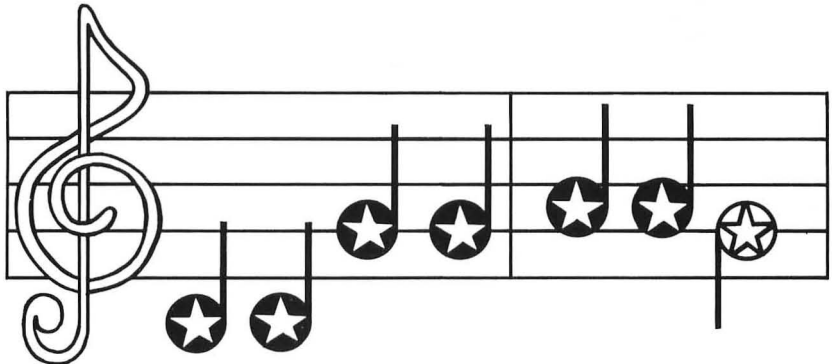
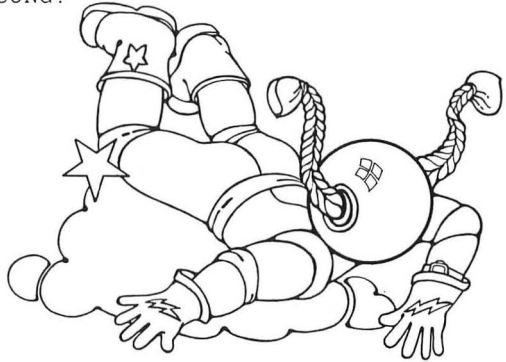
This is how the data lines (250 to 270) were put together. If you want a different song, simply change the data in these lines.

- 200 Reads N,T (in data lines 250 to 270).
- 201 Checks for the flag data (line 280).
- 210 Makes sound N.
- 215–216 Holds the note for the number of beats (T). In this example, each beat will be 80 units long. If you want the song faster, make the number smaller than 80. If you want the song slower, make the number larger than 80.
- 220 Goes back to line 200, and reads the next note (N) and number of beats (T).
- 250–270 Notes and beats.
- 280 Flag data (see line 201).
- 290 Prints a message.

Music from Mother Earth

What's this? They're sending you music from Earth.
They must be testing the sound system!

```
5 GR. 2
200 READ N, T
201 IF N=-1 THEN GOTO 290
210 SOUND 0, N, 10, 8
215 FOR P=1 TO 80*T
216 NEXT P
217 SO. 0, 0, 0, 0
220 GOTO 200
260 DATA 121, 1, 121, 1, 81, 1, 81, 1, 72, 1, 72, 1, 81, 2
270 DATA 91, 1, 91, 1, 96, 1, 96, 1, 108, 1, 108, 1, 121, 2
280 DATA -1, -1
290 PRINT "MY FAVORITE SONG!"
```



Main idea: Using the joystick to play music

Be sure your joystick is plugged into the first jack.

If you plugged into Jack 1 → You have STICK (0)

Jack 2 → You have STICK (1)

Jack 3 → You have STICK (2)

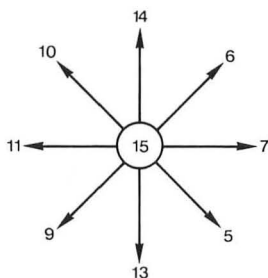
Jack 4 → You have STICK (3)

The program is written for STICK (0).

5 Clears the screen.

6 Prints a message.

10–80 See diagram.



When you press upward, the stick reads “14” and the program says that when the stick reads “14”, sound 121 will be made. Referring to the piano keyboard, you will see that sound 121 is middle C.

Since there are 8 directions, you can get 8 notes, or one full octave. Try playing a scale by moving slowly around the circle.

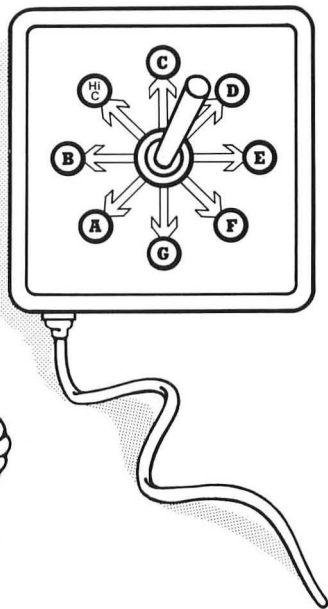
90 When the stick is at rest, there is no sound.

100 Plays the sound.

110 Goes back to line 10 to check for the next sound to play.

Make Your Own Kind of Music

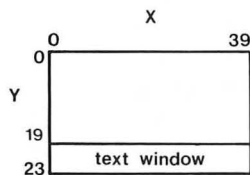
Why not play your own song?
There's a special lever on your
instrument panel. It looks just
like a joystick! You can use it
to make your own music.



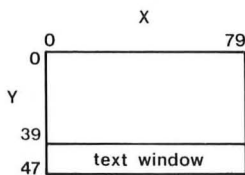
```
5 GR. Ø
6 PRINT "PLAY YOUR OWN MUSIC!"
1Ø IF STICK(Ø)=14 THEN S=121
2Ø IF STICK(Ø)=6 THEN S=1Ø8
3Ø IF STICK(Ø)=7 THEN S=96
4Ø IF STICK(Ø)=5 THEN S=91
5Ø IF STICK(Ø)=13 THEN S=81
6Ø IF STICK(Ø)=9 THEN S=72
7Ø IF STICK(Ø)=11 THEN S=64
8Ø IF STICK(Ø)=1Ø THEN S=6Ø
1ØØ SO.Ø,S,1Ø,8
11Ø GOTO 1Ø
```

Main idea: Beginning graphics

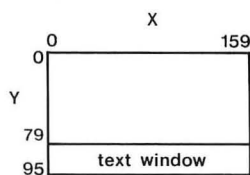
There are 3 graphics modes that take color—Graphics 3, 5, or 7. Each mode has a text window at the bottom and leaves the top of the screen for drawing.



GR.3



GR.5



GR.7

Several abbreviations are useful with graphics:

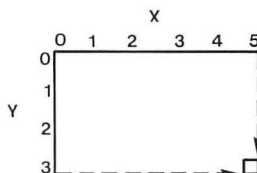
GRAPHICS	GR.
PLOT	PL.
DRAWTO	DR.
COLOR	C.
SETCOLOR	SE.

Color #	Usual color	Can be changed by
Color 1	Gold	SE.0, _____, _____
Color 2	Light green	SE.1, _____, _____
Color 3	Blue (text window)	SE.2, _____, _____
Color 0	Black (background)	SE.4, _____, _____

SETCOLOR, 0 to 15
 SHADE, 0 to 15

Each position on the screen has a name where X meets Y. For example, the point (5,3) is where 5 (the X) meets 3 (the Y). Always put the X first!

PLOT X, Y
DRAWTO X, Y



To see how this program works, enter one line at a time and watch it draw, little by little.

What Is It?

Something scary has appeared
on your screen. What is it?
Is it an alien?

10 GR.5

15 C.1

20 PL.29,4:DR.32,4:DR.32,6:DR.38,6:
DR.38,4:DR.41,4:DR.41,10:DR.39,12:
DR.31,12

30 DR.29,10:DR.29,4

40 PL.34,13:DR.34,21:DR.28,25:DR.28,34:
DR.32,34:DR.32,36:DR.26,36:DR.26,39

50 DR.44,39:DR.44,36:DR.38,36:
DR.38,34:DR.42,34:DR.42,25

60 DR.36,21:DR.36,13

80 C.2

90 SE.1,0,14

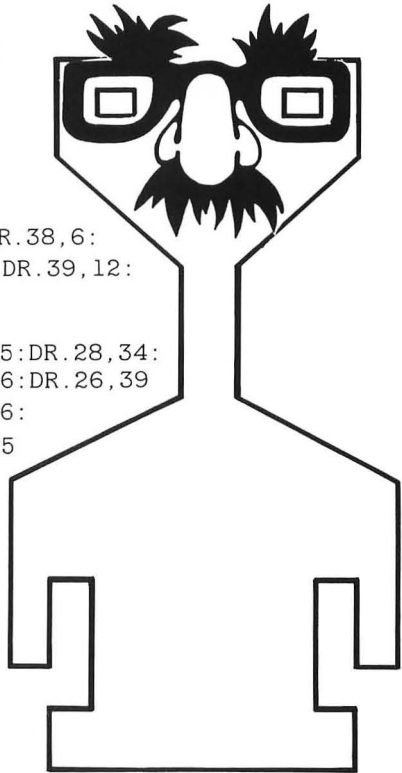
100 PL.30,6:PL.40,6

110 PL.31,6:PL.39,6

120 PL.30,5:PL.39,5

130 PL.32,10:PL.33,11:
DR.37,11:PL.38,10

140 PL.34,7:PL.36,7



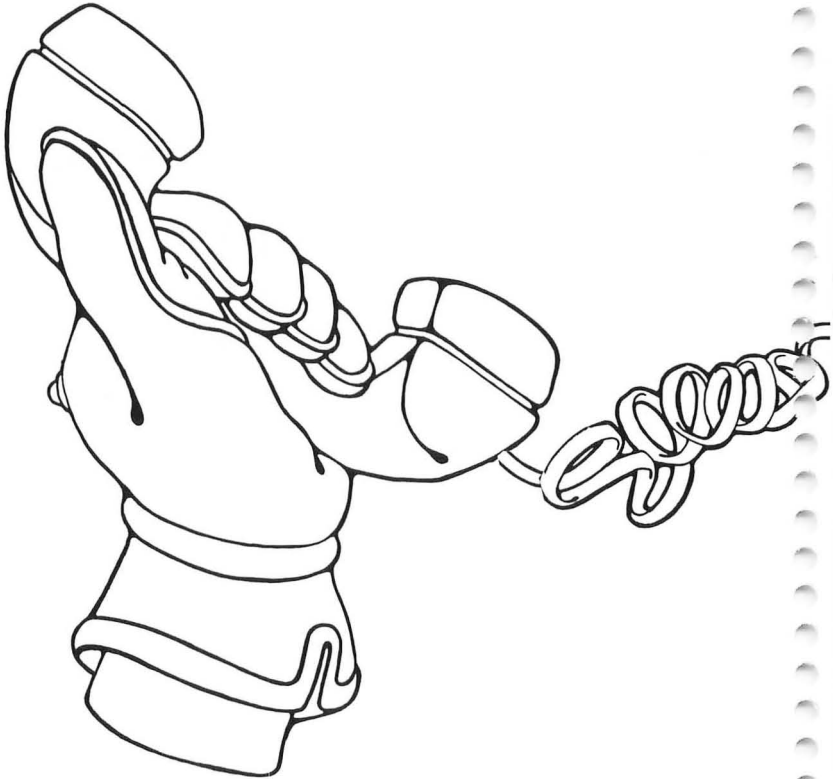
Main ideas: Adding music to graphics
Using the text window

200–270 Plays a song (see explanation in “Music from Mother Earth”).

280–290 Prints a message in the text window.

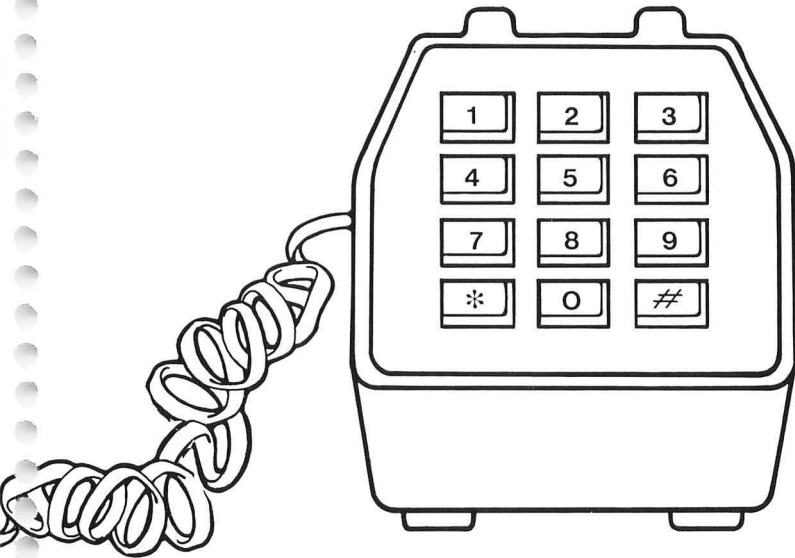
300 Turns off the sound.

310 A dummy line freezes the program so the message stays in the text window.



Who Is It?

Aha! Do you know who it is?



Add these lines to your last program to solve the mystery!

```
200 READ N, T
201 IF N=-1 THEN 280
210 SOUND 0, N, 10, 8
220 FOR B=1 TO 50*T
230 NEXT B
240 GOTO 200
250 DATA 121, 3, 81, 6, 91, 1, 102, 1, 108, 1, 121, 1, 128, 3, 121, 6
260 DATA 144, 3, 72, 6, 81, 1, 91, 1, 96, 1, 108, 1, 114, 3, 108, 6
270 DATA -1, -1
280 PRINT "DON'T BE FOOLED BY HIS GRIN."
290 PRINT "IT'S TEE HEE, UNFRIENDLY ALIEN WHO      NEVER EVER
      PHONES HOME!!!"
300 SOUND 0, 0, 0, 0
310 GOTO 310
```

Main ideas: Random number selection
Incrementing

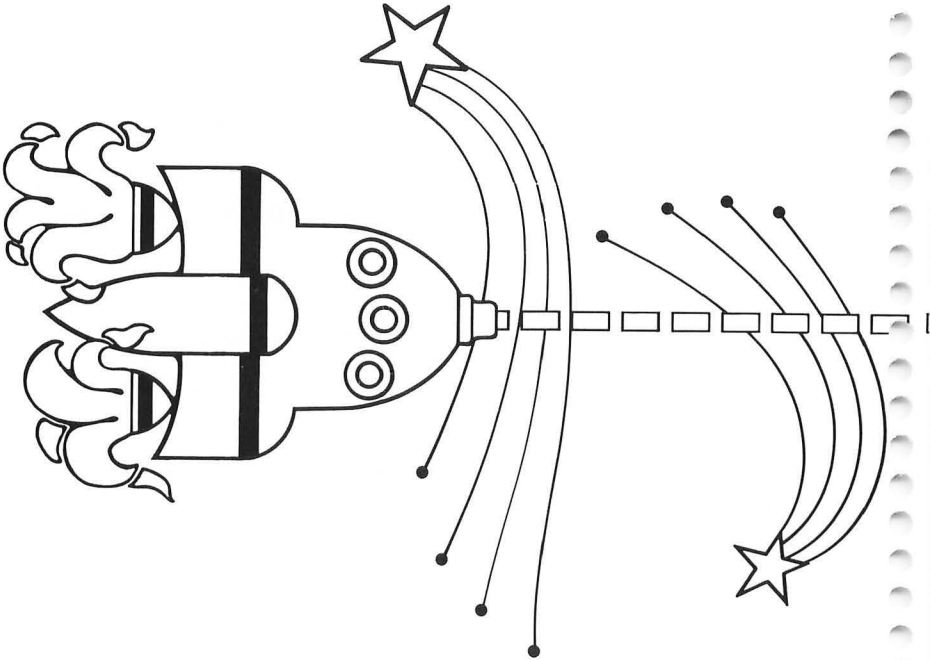
In this program, H = Hits

M = Misses

R = Rounds of ammunition

X = Random number picked by the computer

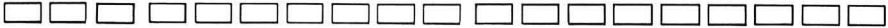
- 10 The counter for hits and misses is set to 0.
- 20–70 The loop that goes through the 500 rounds of ammunition:
 - 30 Selects a random number, 0 or 1.
 - 40 If 0 was selected, it's a hit! Adds 1 to the hits.
 - 50 If 1 was selected, it's a miss! Adds 1 to the misses.
 - 60 Prints the current tally of hits and misses.
 - 70 Goes to line 20 and begins the loop again.
 - 80 The message if the hits total 250 or more.
 - 90 The message if the hits total less than 250.



Hit or Miss?

```
10 H=0:M=0
20 FOR R=1 TO 500
30 X=INT(2*RND(1))
40 IF X=0 THEN H=H+1
50 IF X=1 THEN M=M+1
60 PRINT "HITS ";H,"MISSES ";M
70 NEXT R
80 IF H>=250 THEN PRINT "YOU DID IT!":PRINT "PHONE HOME AND
  TELL THEM ALL ABOUT IT!"
90 IF H<250 THEN PRINT "NO MORE CALLS TO GRANDMA!"
```

Better fire at this awesome creature!
You have 500 rounds of ammunition.
250 hits will knock him out.
Can you do it?
If not, he has promised
to visit Earth and destroy
all the telephones!



Main idea: Using the buzzer in your program

When you type in line 50, push the escape button (ESC) and then push CTRL and 2 at the same time. On the screen, it will say:

```
50 PRINT "␣ "
```

But when the program runs, it makes the buzzer noise!

Some refinements to your program:

1. To change the color of letters in lines 30 or 70, remember:

```
capital letters = gold
lower-case letters = light green
reversed capital letters = blue
reversed lower-case letters = red
```

To get reversed letters, press the ATARI button ⌘.

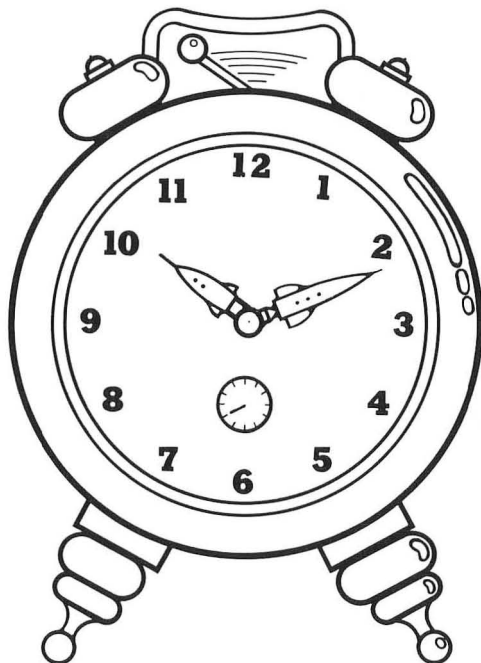
2. Want the buzzer to ring longer? Add:

```
45 FOR A=1 TO 10
80 NEXT A
```

3. For a yellow background on "TIME TO WAKE UP," add:

```
65 SE.4,2,12
```

Nighty Night!

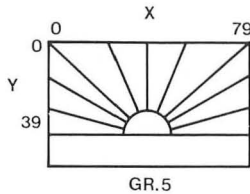


That was hard work! It's time to get a good night's sleep. Before you go to sleep, test your alarm clock by pressing CTRL and 2 at the same time.

```
10 GR.2
20 POS.3,4
30 PRINT #6;"NIGHTY NIGHT!!"
40 FOR P=1 TO 2000:NEXT P
50 PRINT "escape-control-2"
60 POS.2,4
70 PRINT #6;"TIME TO WAKE UP!"
```

Main idea: Using FOR . . . NEXT loops in graphics

- 10 Graphics 5 screen.
- 20 Color 1 is gold.
- 30–60 Goes up the left side.
- 70–100 Goes across the top.
- 110–140 Goes down the right side.



Do you remember that color 0 is black? Here's a nifty addition to your program that draws the program's sunrise first in gold (color 1) and then in black (color 0). When you draw a black sunrise on a black background, the sunrise disappears from view! It then reappears when drawn in gold.

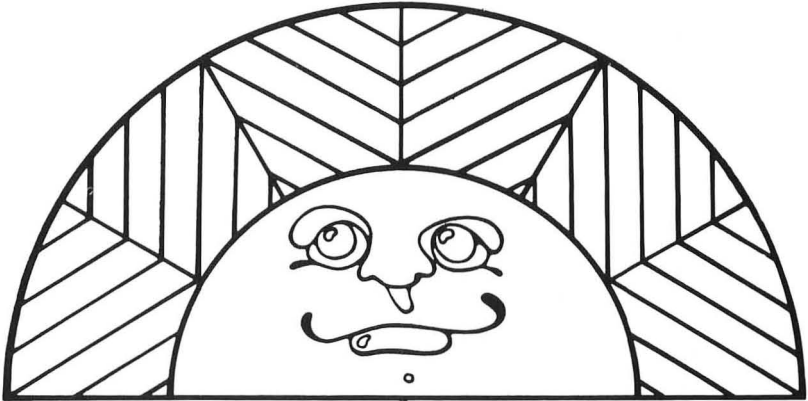
```
15 FOR M = 0 TO 1
20 COLOR M
150 NEXT M
160 GOTO 15
```

Do you remember that color 1 can be changed by using setcolor 0? For colorful sunrises, add these lines:

```
25 C=INT(16*RND(1))
26 SETCOLOR 0,C,6
```

Line 25 selects a number from 0 to 15 for the setcolor. The "6" in line 26 calls for a medium shade of that setcolor.

Good Morning!



Each time you circle Earth, you see a beautiful new sunrise.
Take a look!

```
10 GR.5  
20 C.1  
30 FOR Y=39 TO 0 STEP -13  
40 PL.40,39  
50 DR.0,Y  
60 NEXT Y  
70 FOR X=20 TO 79 STEP 20  
80 PL.40,39  
90 DR.X,0  
100 NEXT X  
110 FOR Y=0 TO 39 STEP 13  
120 PL.40,39  
130 DR.79,Y  
140 NEXT Y
```

Main ideas: Using OPEN-GET

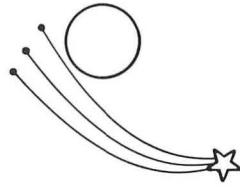
Drawing without joysticks

Input from the keyboard using ASCII numbers

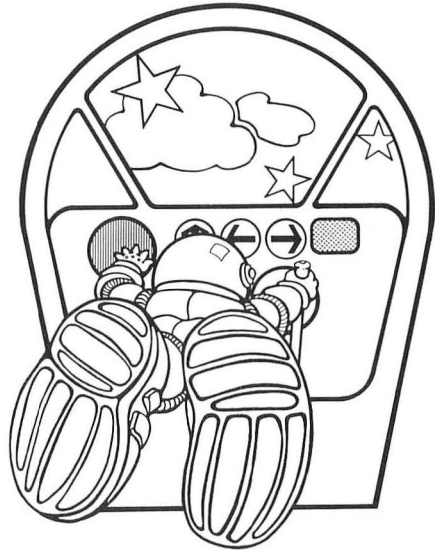
- 10 Graphics 5. The "+16" removes the text window.
- 15 Color 1 is gold.
- 20-60 Makes 100 random dots.
- 70-100 Sets a red dot at the bottom of the screen.
 - 120 Opens the keyboard for the "GET" statement in line 130.
 - 130 The N is the ASCII number of the key pressed.
 - 140 45 is the ASCII code for "-", which is on the ↑ key.
 - 150 61 is the ASCII code for "=", which is on the ↓ key.
 - 160 42 is the ASCII code for "*", which is on the → key.
 - 170 43 is the ASCII code for "+", which is on the ← key.
- 175-176 Keeps errors away if you are about to go off the screen.
 - 180 Plots your path through the "meteors."
 - 185 If you get to the top of the screen, goes to line 200.
 - 190 Continues the loop to plot your path.
 - 200 Prints a message.

Meteors!

It's a meteor shower.
Can you steer your way
through it? Use the arrows
to plot your way. Try to
make it to the top of the
screen! Good Luck!



```
10 GR.5+16
15 C.1
19 REM METEOR SHOWER
20 FOR M=1 TO 200
30 X=INT(80*RND(1))
40 Y=INT(48*RND(1))
50 PL.X,Y
60 NEXT M
70 REM RED DOT
80 X=40:Y=47
90 C.2
95 SE.1,4,4
100 PL.X,Y
119 REM STEERING
120 OPEN #1,4,0,"K"
130 GET #1,N
140 IF N=45 THEN Y=Y-1
150 IF N=61 THEN Y=Y+1
160 IF N=42 THEN X=X+1
170 IF N=43 THEN X=X-1
175 IF X<0 THEN X=0
176 IF X>79 THEN X=79
180 PL.X,Y
185 IF Y=0 THEN 200
190 GOTO 130
200 PRINT "YAY! YOU MADE IT!"
```



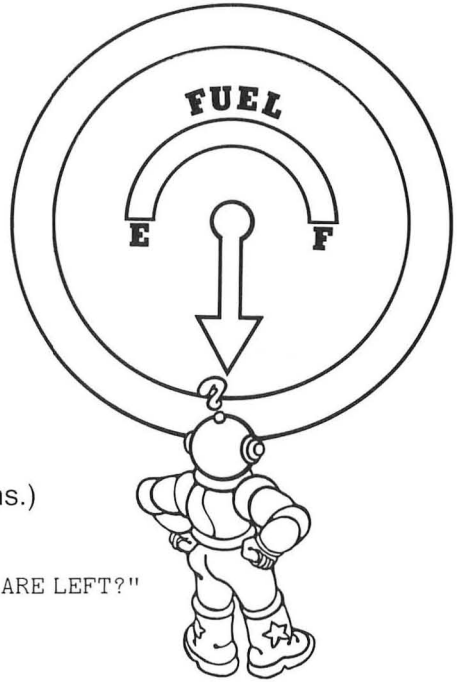
Main idea: A guessing game

- 5 Clears the screen.
- 10 Prints a message.
- 20 Selects a random integer (a whole number), 0 to 99.
- 30 Allows you to guess the number.
- 40–60 Evaluates your guess.
- 70 Goes back for another guess.
- 100 Prints “RIGHT!”

The following section evaluates the amount of fuel that is left and prints an appropriate message.

- 100 Graphics 2 is for big letters.
- 110 Positions the message.
- 120 If the fuel is less than 30 gallons, this message is printed.
- 130 If the fuel is 30 to 59 gallons, this message is printed.
- 140 If the fuel is 60 gallons or more, this message is printed.

How Much Fuel?



Oh no! The fuel gauge is broken! Guess how much fuel you have left. (A full tank holds 100 gallons.)

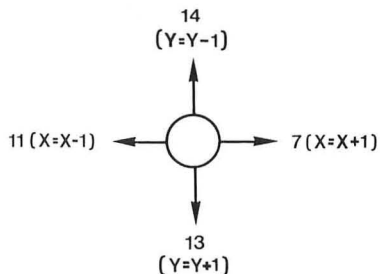
```
5 GR. 0
10 PRINT "HOW MANY GALLONS ARE LEFT?"
20 F=INT(100*RND(1))
30 INPUT G
40 IF G=F THEN 100
50 IF G<F THEN PRINT "YOU HAVE MORE THAN THAT!"
60 IF G>F THEN PRINT "YOU DON'T HAVE THAT MUCH!"
70 GOTO 30
100 PRINT "RIGHT!"
```

By the way, you need at least 30 gallons to land safely!

```
100 GR. 2
110 POS. 4, 5
120 IF F<30 THEN PRINT #6; "BAD NEWS!": END
130 IF F<60 THEN PRINT #6; "SHOULD BE OK!": END
140 PRINT #6; "EXCELLENT!"
```

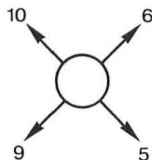
Main idea: Drawing with joysticks

The diagram shows the joystick readings when it is pressed in any of the directions indicated. Refer to the diagram to help you understand how the program works.



- 10 Graphics 7 will be used. The +16 removes the text window.
- 20 The starting point of the drawing (mid-screen).
- 25 Initially gold (color 1) is its usual color.
- 30 Plots the current location (X,Y).
- 40 If you press toward the right, the joystick reads 7, and 1 is added to the X coordinate.
- 50 If you press left, 1 is subtracted from the X coordinate.
- 60 If you press down, 1 is added to the Y coordinate.
- 70 If you press up, 1 is subtracted from the Y coordinate.
- 170 Goes back to line 25.

When you draw diagonally, both the X coordinate and the Y coordinate are changed. See if you can understand what happens in lines 80 to 110.



To prevent errors, you cannot have an X number larger than 159 or a Y number larger than 95. Neither X nor Y can be less than 0. (Refer to Graphics 7 Sheet in your Appendix.) Lines 120 to 150 keep the X and Y numbers from getting too high or too low.

As for the color, refresh your memory about color and setcolor by referring to the program called "What Is It?" In that program, you learned that setcolor 0 alters color 1, the basic gold color that we are currently using. You also learned that setcolor 4 controls the background color.

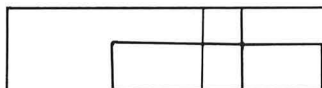
In this program, each time you press the "trigger" on your joystick, STRIG(0) becomes 0 and the number C goes up 1. When the program goes back to draw your current location, it must execute lines 26 and 27, which change setcolor 0 and setcolor 4 as indicated. If you recall, the setcolor colors range from 0 to 15. The computer resolves numbers higher than 15. For example, 16 is the same as 0, 17 is the same as 1, etc.

Space Sketch

As you near your destination, you see something strange. Sketch what you see. NASA will want to study your drawing, so do the best you can do!

```

10 GR.7+16
20 X=80:Y=48
25 COLOR 1
30 PL.X,Y
40 IF STICK(0)=7 THEN X=X+1
50 IF STICK(0)=11 THEN X=X-1
60 IF STICK(0)=13 THEN Y=Y+1
70 IF STICK(0)=14 THEN Y=Y-1
170 GOTO 25
  
```



Tired of errors?
Add these lines:

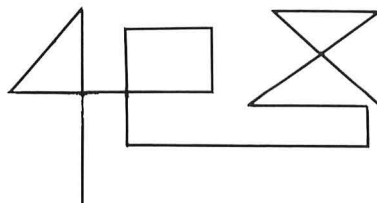
```

120 IF X>159 THEN X=159
130 IF X<0 THEN X=0
140 IF Y>95 THEN Y=95
150 IF Y<0 THEN Y=0
  
```

Want to draw diagonally, too?

```

80 IF STICK(0)=6 THEN X=X+1:Y=Y-1
90 IF STICK(0)=5 THEN X=X+1:Y=Y+1
100 IF STICK(0)=10 THEN X=X-1:Y=Y-1
110 IF STICK(0)=9 THEN X=X-1:Y=Y+1
  
```



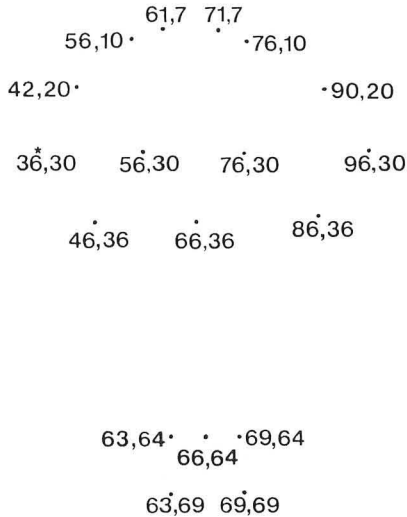
How about some color?

```

26 SETCOLOR 0,C,6
27 SETCOLOR 4,C+5,10
160 IF STRIG(0)=0 THEN C=C+1
  
```

Main ideas: “Counting” sounds
Drawing a picture
Using the SELECT key

- 4 Clears the screen.
- 10–30 Makes sounds 1 to 200 counting by fives. For a change, try STEP 3 or STEP 7 or some other step number.
- 25 Pauses to hold the note.
- 100–180 Draws the parachute. Start with line 130 and find the dots in the diagram. Connect the dots so you can see how the parachute is drawn. If you are feeling super ambitious, use the Graphics 7 Sheet in the Appendix to try to plot the parachute.



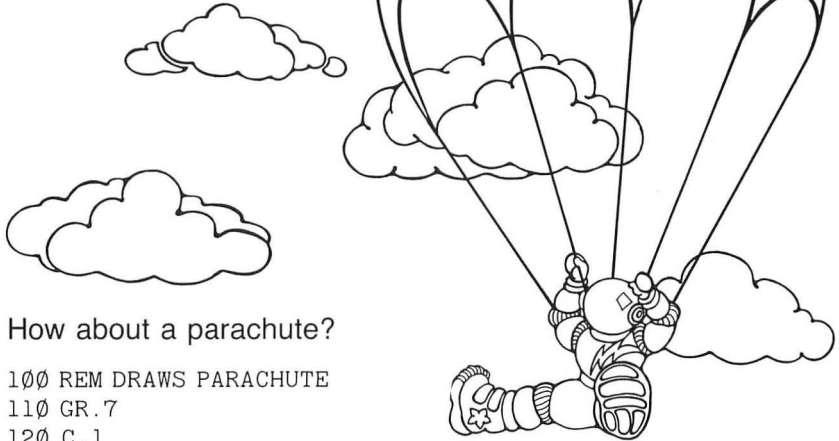
This section will make a game out of the program:

- 5 Prints the direction at the beginning of the program.
- 26–27 When you press “SELECT” the number stored at PEEK (53279) becomes 5.
- 40–210 Checks the sound that the program is up to when the “SELECT” button is pressed. More than 150 is too late! Less than 100 is too early! Otherwise, it’s a perfect splashdown!
- 190 Skips a line in the text window.

Happy Landing!

You are descending to Earth.

```
4 GR. 0
10 FOR N=1 TO 200 STEP 5
20 SO. 0, N, 8, 8
25 FOR P=1 TO 50: NEXT P
30 NEXT N
```



How about a parachute?

```
100 REM DRAWS PARACHUTE
110 GR. 7
120 C. 1
130 PL. 36, 30: DR. 46, 36: DR. 56, 30: DR. 66, 36: DR. 76, 30: DR. 86,
    36: DR. 96, 30
140 DR. 90, 20: DR. 76, 10: DR. 71, 7: DR. 61, 7: DR. 56, 10: DR. 42,
    20: DR. 36, 30
150 PL. 66, 64: DR. 66, 36
160 PL. 66, 64: DR. 86, 36
170 PL. 66, 64: DR. 46, 36
180 PL. 63, 64: DR. 69, 64: DR. 69, 69: DR. 63, 69: DR. 63, 64
220 SOUND 0, 0, 0, 0
230 GOTO 230
```

Did you release the parachute on time?

```
5 PRINT "PRESS SELECT FOR PARACHUTE!"
26 PK=PEEK(53279)
27 IF PK=5 THEN GOTO 40
40 IF N>150 THEN PRINT "TOO LATE!":END
50 IF N<100 THEN PRINT "TOO EARLY!":END
190 PRINT
200 PRINT " A PERFECT SPLASHDOWN!"
210 PRINT " WELCOME BACK TO EARTH!"
```

Main ideas: Simple word processing
Substituting variables in a text

5 Dimensions all string variables used.

6 Clears the screen.

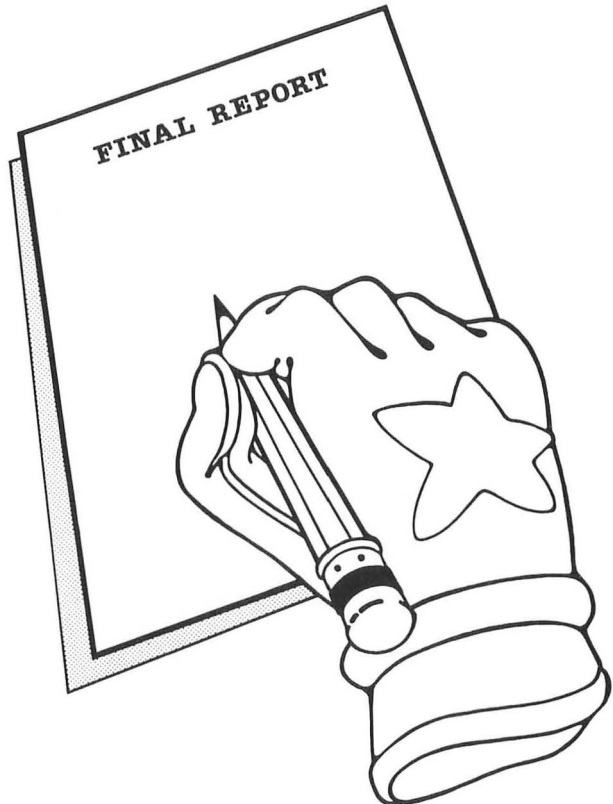
10–70 A chance to input variables.

110 Clears the screen.

120–170 Prints the report using the variables input in lines 10 to 70.

Note the use of semicolons and spaces in getting the report to print out correctly. Semicolons make the next word printed appear immediately after the last word printed. You must put the spaces in where you want them.

Also note the difference between string variables and numeric variables: In lines 10 to 60, you put in words—hence N\$, G\$, PL\$, L\$, D\$ (string variables). In line 70, you put in a number—hence N (no dollar sign, since N is a numeric variable).



Final Report

It's time to evaluate your mission. The General wants your report as soon as possible.

```
5 DIM N$(20), PL$(10), L$(15), D$(10), G$(20)
6 GR. 0
10 PRINT "YOUR NAME";
20 INPUT N$
25 PRINT "NAME OF GENERAL IS GEN. ";
30 INPUT G$
35 PRINT "PLANET VISITED";
40 INPUT PL$
45 PRINT "LENGTH OF TRIP";
50 INPUT L$
55 PRINT "DAY OF WEEK";
60 INPUT D$
65 PRINT "A NUMBER";
70 INPUT N
100 REM PRINT REPORT
110 GR. 0
120 PRINT "REPORT OF "; N$
130 PRINT "TO GEN. "; G$
140 PRINT
150 PRINT "MY MISSION TO "; PL$; " IS COMPLETED."
160 PRINT "THE MISSION LASTED "; L$
170 PRINT "I WILL SEND YOU A "; N; " PAGE REPORT NEXT "; D$
```

Main idea: Moving an image across the screen to create a “race”

In this program, U = United States' position in the race

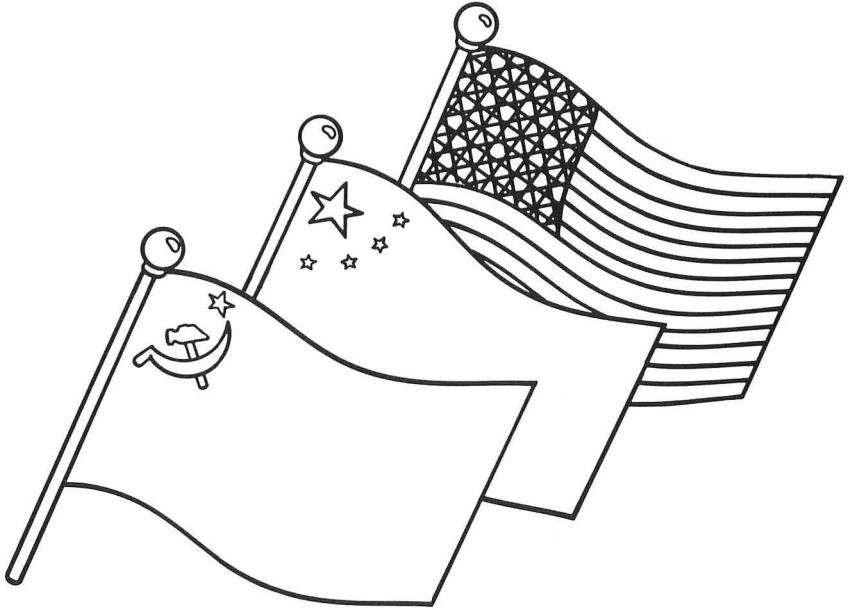
R = Russia's position in the race

C = China's position in the race

- 10 Clears the screen.
- 11 Positions and prints the title of the race.
- 12 Gets rid of the cursor (see what happens if you leave out this line).
- 17 Lines “runners” up at the starting point (all start at position 7).
- 20–40 Positions and prints the names of the “runners.”
- 50 Picks a random integer, 1 to 3.
- 51 Each line ends at box 39. If any runner gets to box 40, the race is over!
- 60 If the computer picked a 1 in line 50, it prints a “#” for the U.S. and adds 1 to its position for the next time.
- 70 If it picked a 2 in line 50, it prints a “*” for Russia and adds 1 to its position for the next time.
- 80 If it picked a 3 in line 50, it prints an “@” for China and adds 1 to its position for the next time.
- 90 Goes back to line 50 to pick another number to advance a runner.
- 200 If someone is at box 40 and it's not the U.S., the computer prints the “too bad” message.
- 205 If someone is at box 40 and it *is* the U.S., the computer prints the congratulatory message.
- 210 A dummy line to avoid the “READY” message.

The Race for Space

Your voyage is over, Captain. Have you done the very best for your country? Find out who has won the race for space.



```
10 GR. 0
11 POS. 9, 1: PRINT "THE RACE FOR SPACE"
12 POKE 752, 1
17 U=7:R=7:C=7
20 POS. 0, 6: PRINT "U. S. A. "
30 POS. 0, 10: PRINT "USSR"
40 POS. 0, 14: PRINT "CHINA"
50 X=INT(3*RND(1))+1
51 IF U=40 OR R=40 OR C=40 THEN GOTO 200
60 IF X=1 THEN POS. U, 6: PRINT "#":U=U+1
70 IF X=2 THEN POS. R, 10: PRINT "*":R=R+1
80 IF X=3 THEN POS. C, 14: PRINT "@":C=C+1
90 GOTO 50
200 IF U<40 THEN POS. 0, 20: PRINT "TOO BAD, CAPTAIN! YOU MUST
    TRY HARDER!"
205 IF U=40 THEN POS. 5, 20: PRINT "CONGRATULATIONS, CAPTAIN!"
210 GOTO 210
```

Main idea: Using ASCII codes, CHR\$(N) and ASC(A\$)

- 5 Clears the screen.
- 6 Gets rid of the cursor (makes it invisible).
- 10 Gets the cursor in position for the secret message.
- 15 Reads the number in the data line.
- 16 Checks for the flag data “- 1”, which is at the end of the data line.
- 20 Prints the symbol that belongs to the number N.
- 30 Goes to line 15 to read the next number.
- 50 Data. The numbers stand for symbols (see chart).
- 100 A dummy line freezes the image and prevents a “READY” message.

This section works in reverse. You give the computer a letter and it gives you the code number.

- 5 Clears the screen.
- 10 Dimensions A\$ at 1 letter.
- 20 Prints the question.
- 30 Allows you to tell the letter, number, or symbol you want.
- 40 A is the number that is the ASCII code for the input in line 30.
- 50 Prints the code number.
- 60 Goes to line 20 to ask for the next character.

Break the Code

33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

The General has sent you this final message. Run the program to break the code.

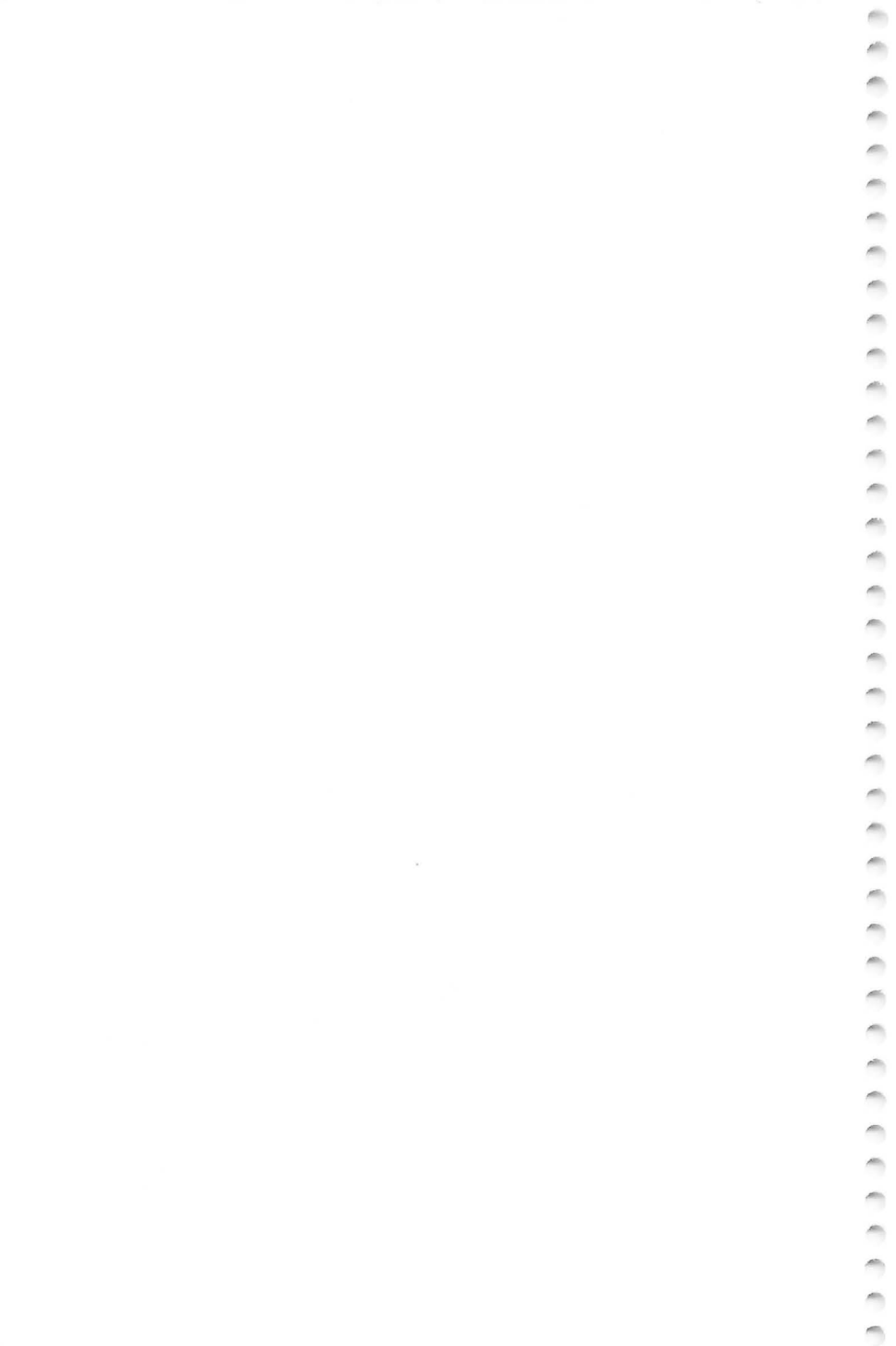
```
5 GR.Ø
6 POKE 752,1
1Ø POS.1Ø,8
15 READ N
16 IF N=-1 THEN 1ØØ
2Ø PRINT CHR$(N);
3Ø GOTO 15
5Ø DATA 79,86,69,82,32,65,78,68,32,79,85,84,33,-1
1ØØ GOTO 1ØØ
```

To write a return message, put your own numbers into line 50. Use the chart or run this program to get the numbers you need.

```
5 GR.Ø
1Ø DIM A$(1)
2Ø PRINT "WHAT CHARACTER DO YOU WANT TO KNOW?"
3Ø INPUT A$
4Ø A=ASC(A$)
5Ø PRINT "ITS CODE IS ";A
6Ø GOTO 2Ø
```

THE END

84 72 69 32 69 78 68



APPENDIX

Abbreviations, 45

Control Graphics Keyboard, 46

Graphics Sheets, 47–52

Graphics 0, 47

Graphics 1, 48

Graphics 2, 49

Graphics 3, 50

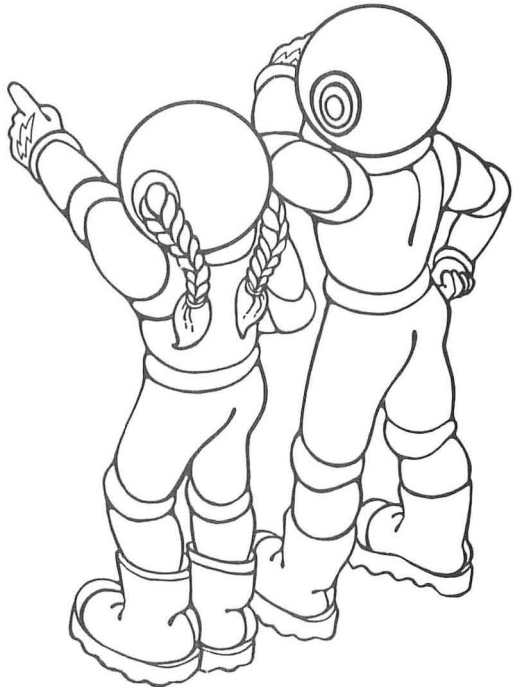
Graphics 5, 51

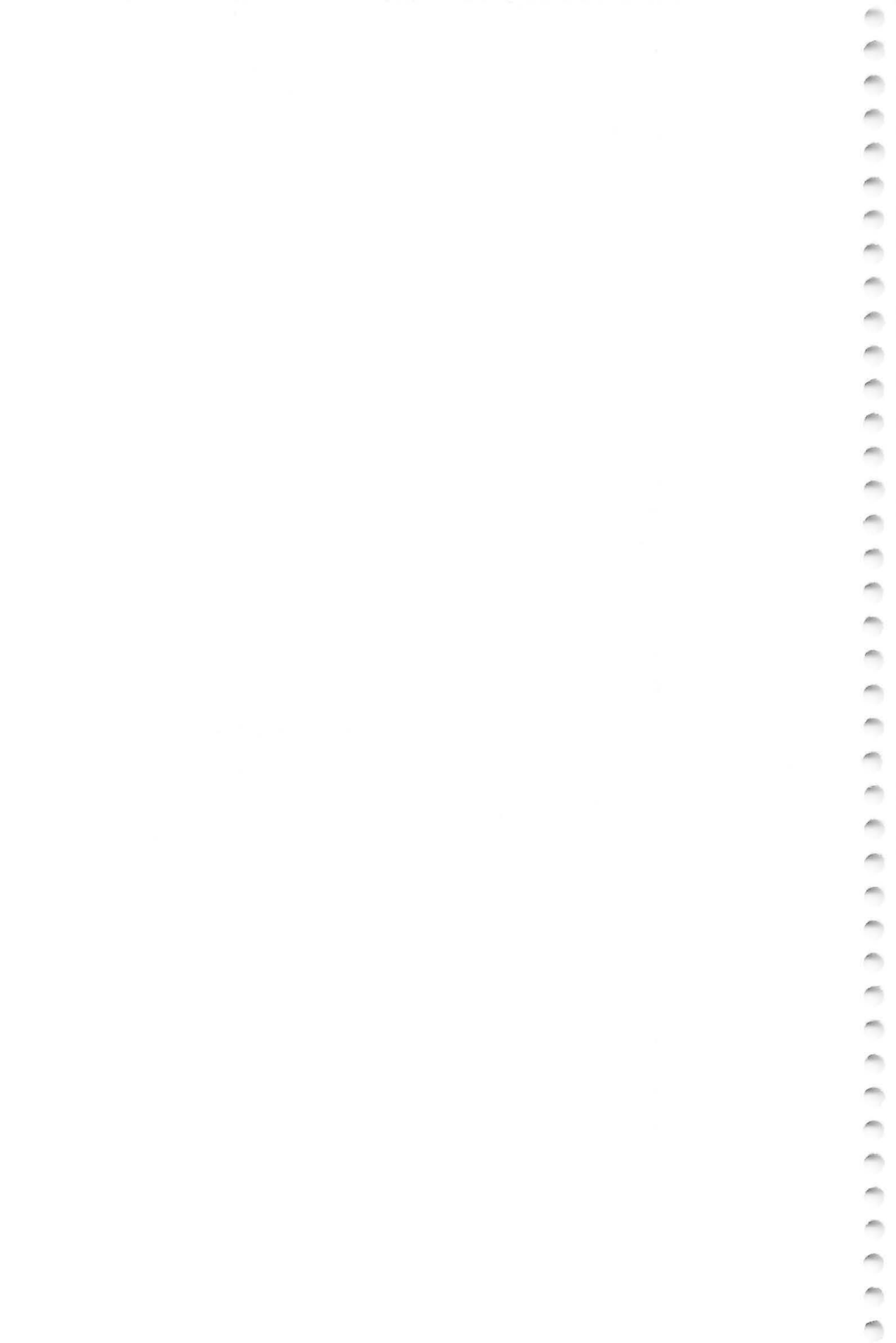
Graphics 7, 52

Musical Notes, 53

Saving Programs on Cassette, 54

Setcolor Chart, 55



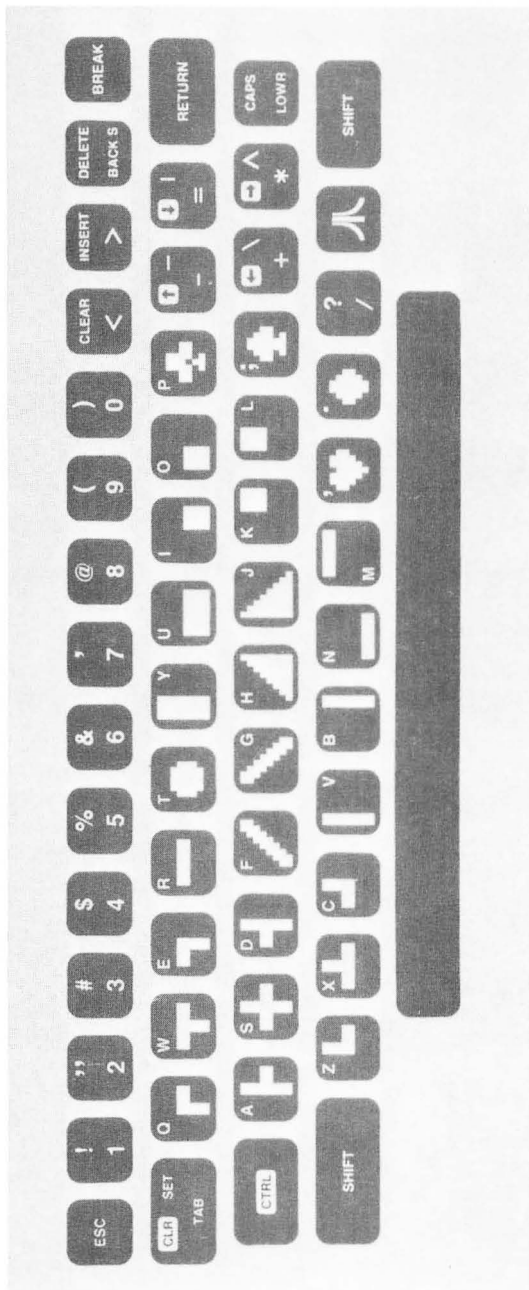


Abbreviations

This is a partial list of the abbreviations available. Use them, but remember what they stand for!

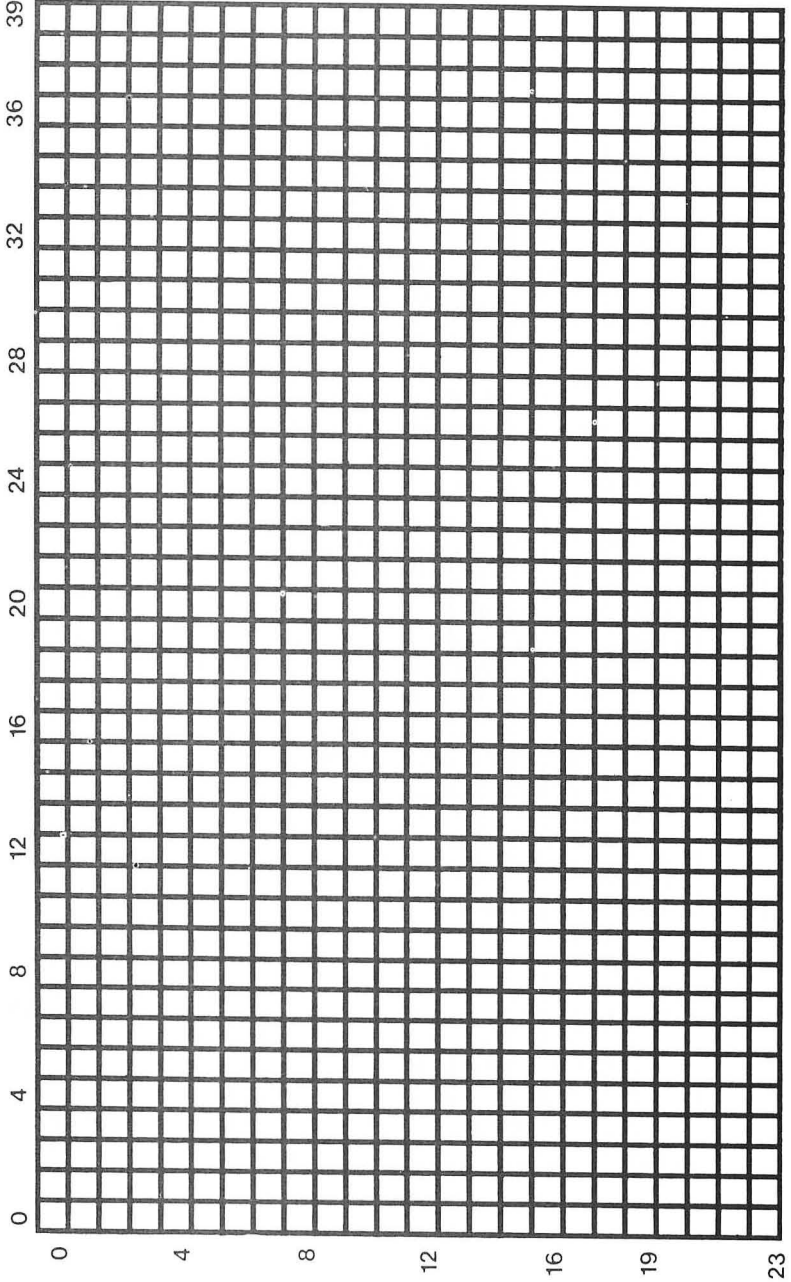
C. COLOR
D. DATA
DR. DRAWTO
F. FOR
G. GOTO
GOS. GOSUB
GR. GRAPHICS
I. INPUT
L. LIST
LOC. LOCATE
N. NEXT
PL. PLOT
POS. POSITION
PR. PRINT
? PRINT
RES. RESTORE
RET. RETURN
SE. SETCOLOR
SO. SOUND

Control Graphics Keyboard

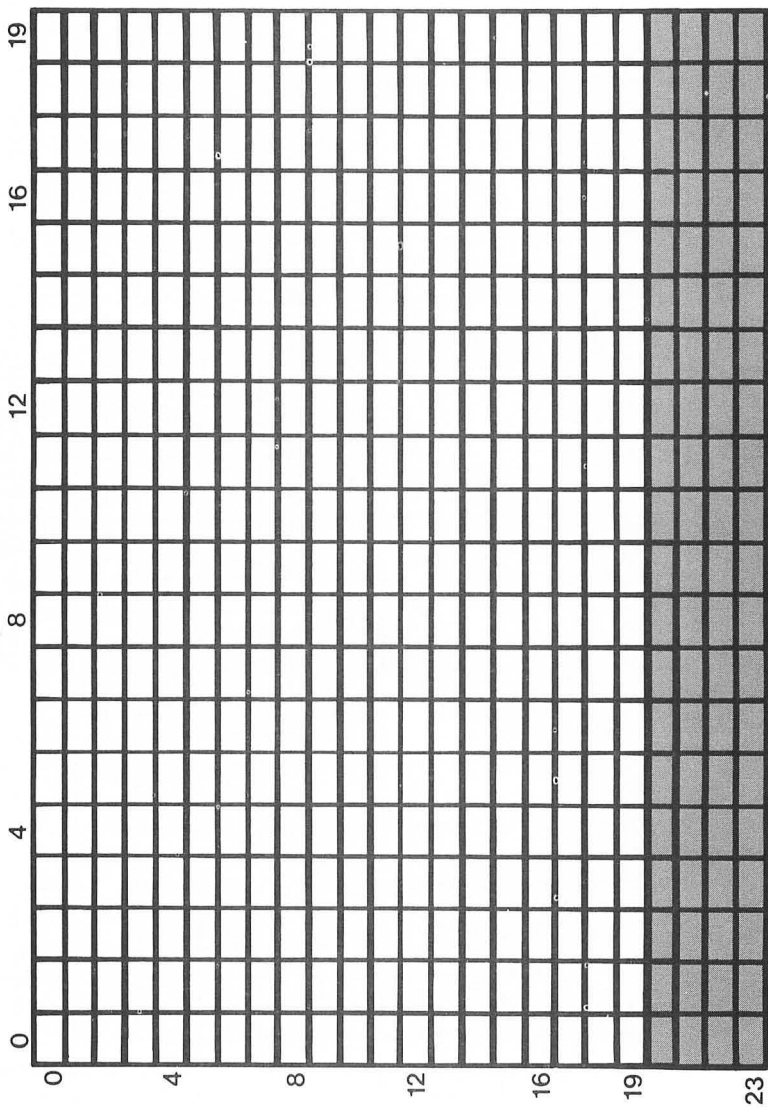


Photograph reprinted by permission of Atari, Inc. © 1980

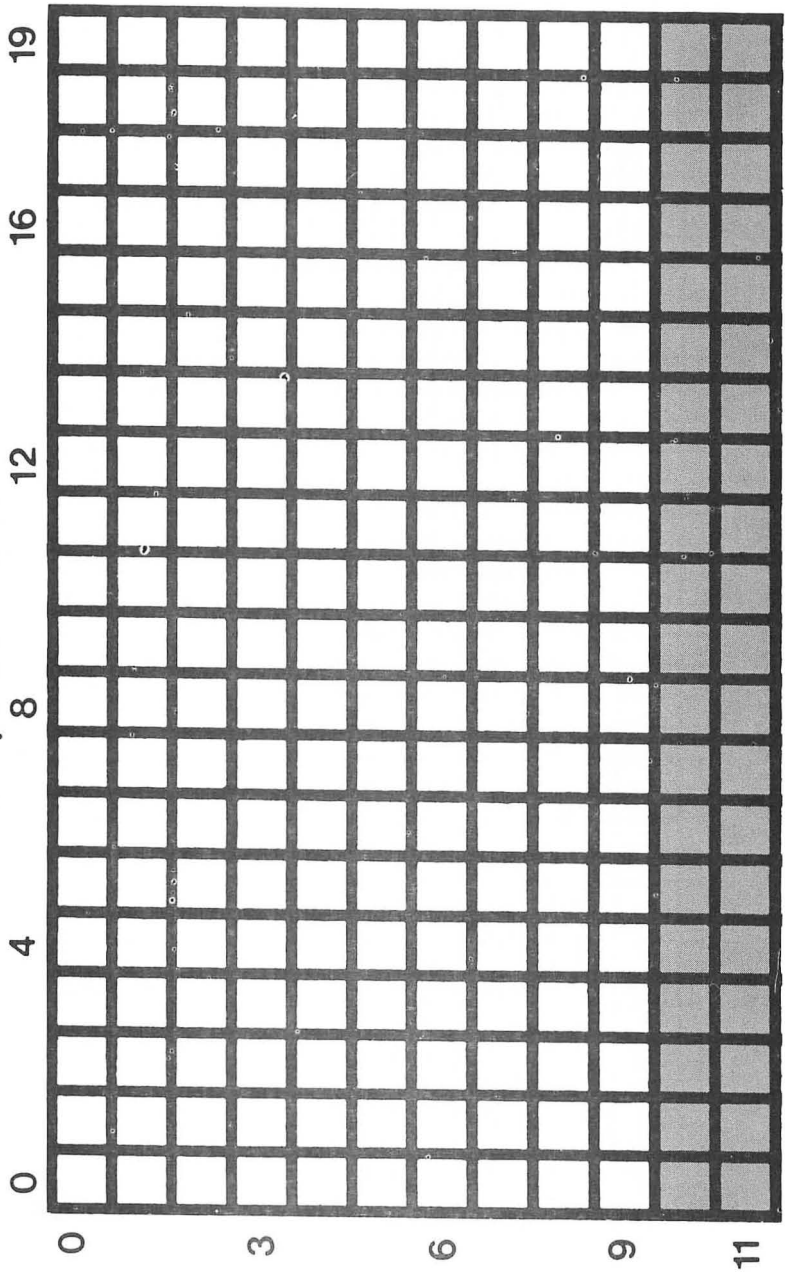
Graphics 0 Sheet



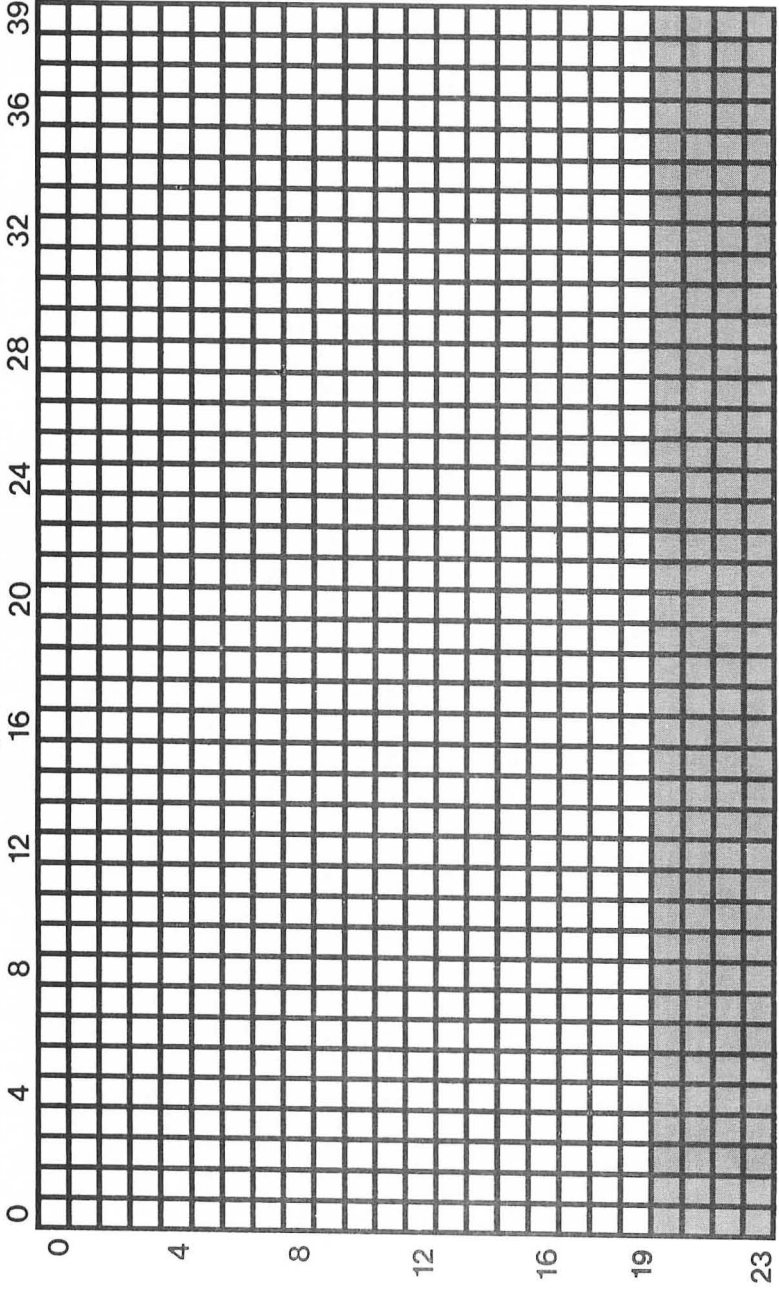
Graphics 1 Sheet



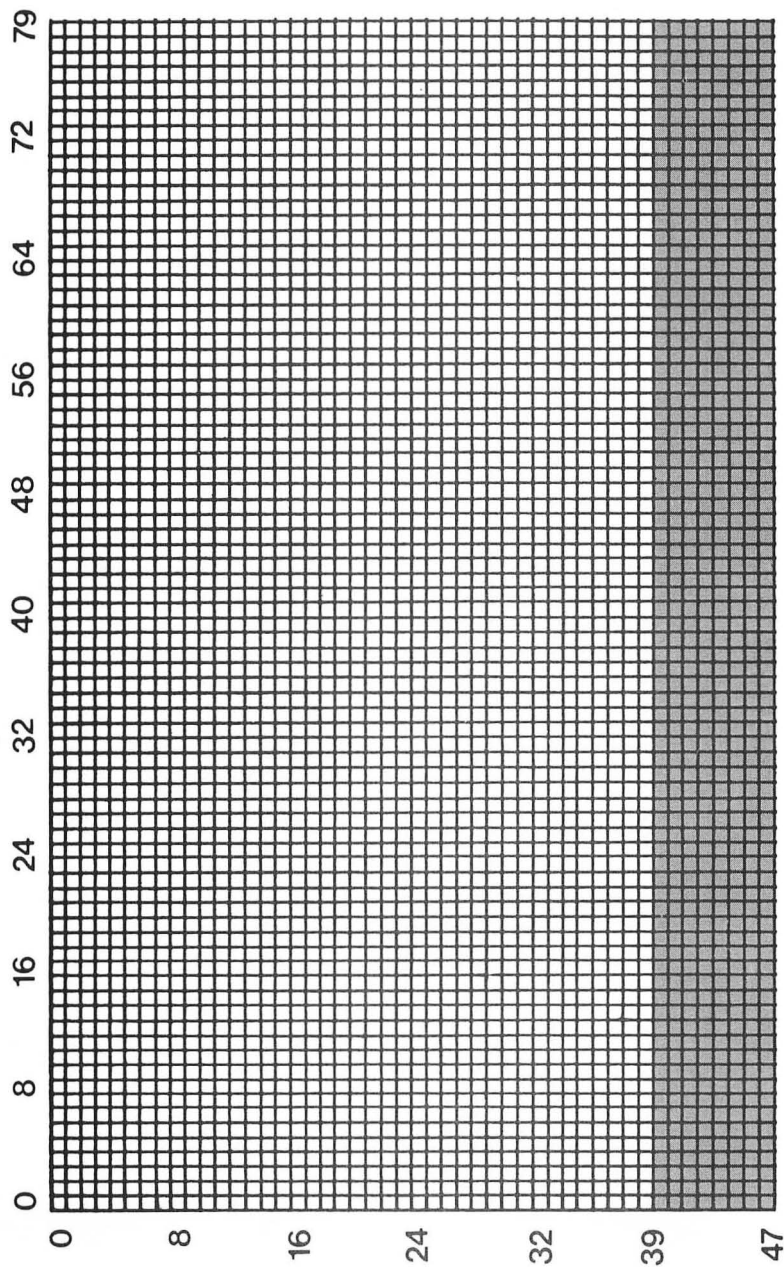
Graphics 2 Sheet



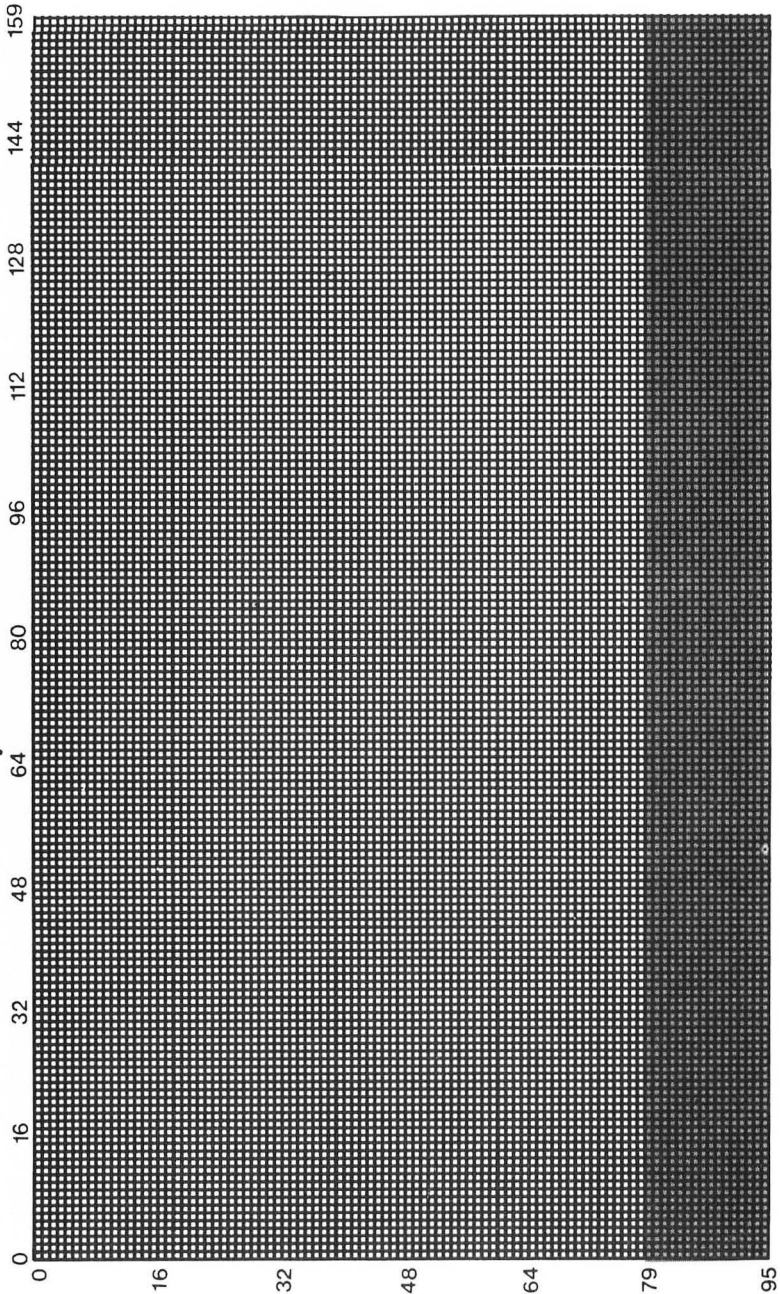
Graphics 3 Sheet



Graphics 5 Sheet



Graphics 7 Sheet



Saving Programs on Cassette

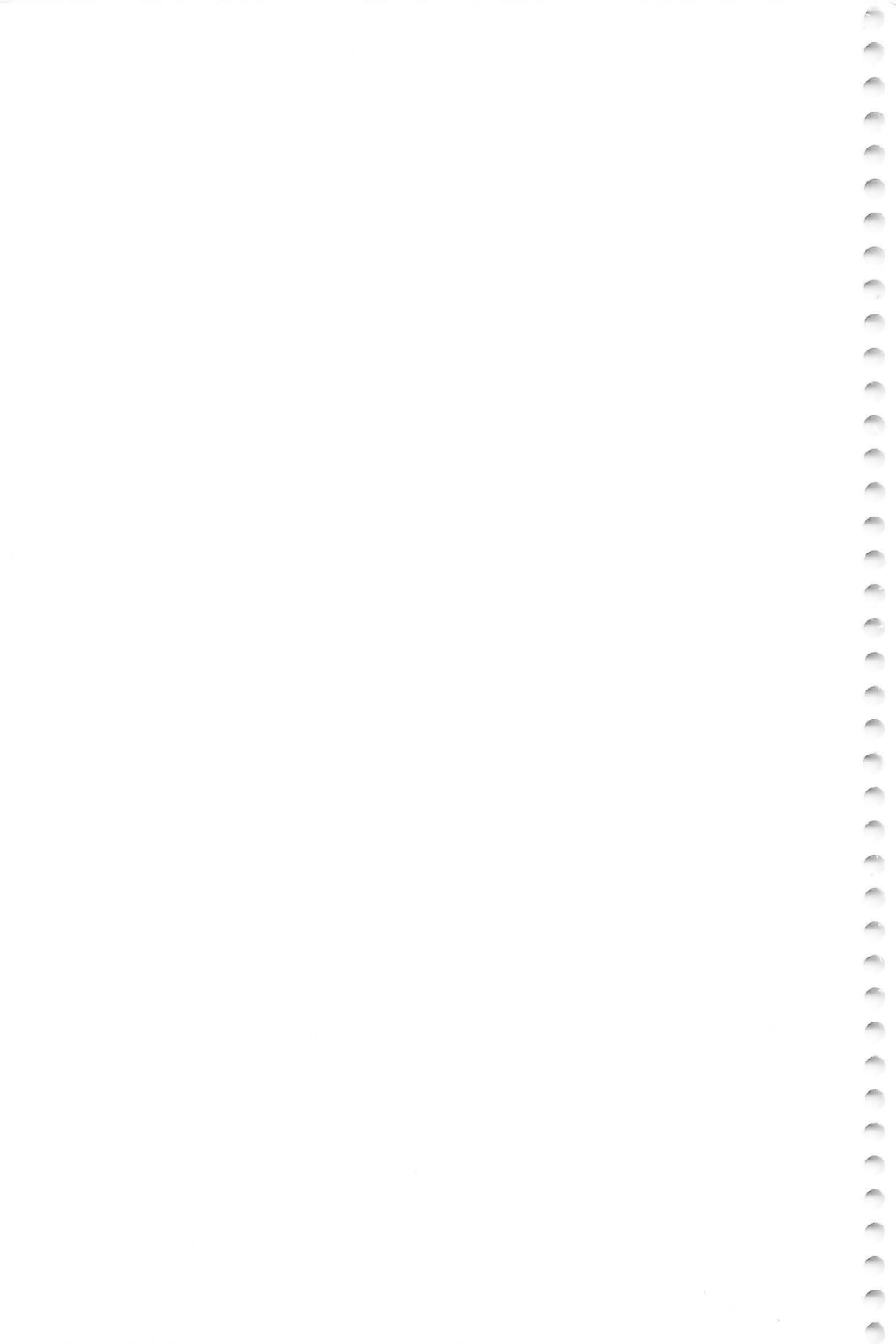
Are you having problems? I find this procedure to be more reliable than the usual CSAVE-CLOAD. It takes a little longer, but it saves a lot of time and aggravation in the long run.

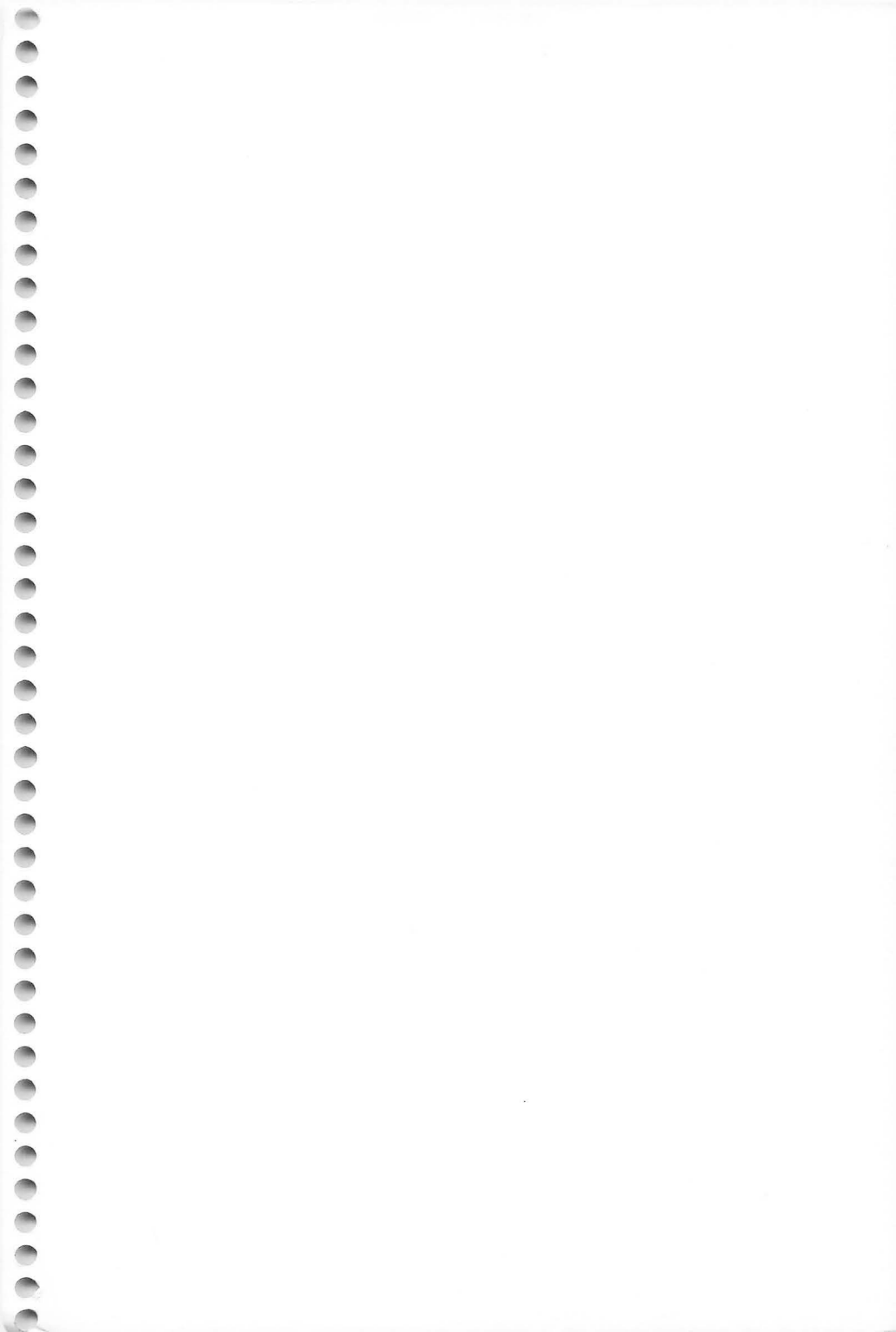
1. Position the tape at a new location. (Don't tape over old programs!) Remember the number.
2. Type LIST "C:" and press RETURN.
3. The computer will buzz twice. Make sure two buttons on the cassette recorder are down, PLAY and RECORD. Then press RETURN again.
4. When the program is saved, the screen will say READY.
5. To verify that your program is saved, reposition the tape and type ENTER "C:".
6. The computer will buzz once. Make sure that the PLAY button is down. Press RETURN.
7. If you get no errors, the program is properly saved. If you do get an error, don't panic! The original program is still in memory. Find a new place on the tape and try to save it again, beginning with step 2 above.
8. Save your money for a disk drive.

Setcolor Chart

Color it in to help you out!

0	black
1	rust
2	red-orange
3	dark orange
4	red
5	dark lavender
6	cobalt blue
7	ultramarine blue
8	medium blue
9	dark blue
10	blue-gray
11	olive green
12	medium green
13	dark green
14	orange-green
15	orange





\$6.95

CLAIRE BAILEY PASSANTINO

ITTY BITTY BYTES OF SPACE



SPACE—an endless source of fascination for us all. Children, however, have the imagination to project themselves into orbit as effortlessly as they learn to talk. Now, ITTY BITTY BYTES OF SPACE encourages your child to explore the magic of space with the help of your home computer. Young readers are presented with bite-size programs and activities designed to excite their eyes, ears, and minds. Each program is accompanied by programming hints and suggestions for parents and teachers. Help your children learn, grow, and play—the computer way!

Table of Contents: A Note to Parents and Teachers • Passing the Test • Gravity • Test Your Spacecraft • Take Off • Count the Stars • A Stellar Sight • Music from Mother Earth • Make Your Own Kind of Music • What Is It? • Who Is It? • Hit or Miss? • Nighty Night! • Good Morning! • Meteors • How Much Fuel? • Space Sketch • Happy Landing! • Final Report • The Race for Space • Break the Code

Other **Itty Bitty Bytes** Books: Matilda the Computer Cat
School Days

For more information on Creative Pastimes Books write to:
Reston Computer Group
11480 Sunset Hills Road
Reston, VA 22090

illustration by Bethann Thornburgh



Reston Publishing Company, Inc.
A Prentice-Hall Company
Reston, Virginia

0-8359-3316-4