An ATARI® in

the Classroom
An ATARI in the Classroom Activity Workbook is part of a three-book set. As a workbook, it is not eligible to be catalogued by the Library of Congress. The following data applies to An ATARI for Kids, the principle text in the set.

Library of Congress Cataloging in Publication Data

Boren, Sharon, 1956--
An Atari for kids.

Includes index.
Summary: Teaches beginning programmers how to program the Atari or other microcomputers in the BASIC computer language.
1. Atari computer — Programming — Juvenile literature. 2. BASIC (Computer program language) — Juvenile literature.
QA76.8.A82B67 1984 001.64'2 83-25269

ISBN 0-88056-123-8 (pbk.)
©1984 by dilithium Press. All rights reserved.

No part of this book may be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system without permission in writing from the publisher, with the following exceptions: any material may be copied or transcribed for the nonprofit use of the purchaser, and material (not to exceed 300 words and one figure) may be quoted in published reviews of this book.

Where necessary, permission is granted by the copyright owner for libraries and others registered with the Copyright Clearance Center (CCC) to photocopy any material herein for a base fee of $1.00 and an additional fee of $0.20 per page. Payments should be sent directly to the Copyright Clearance Center, 21 Congress Street, Salem, Massachusetts 01970.

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging in Publication Data

Printed in the United States of America

dilithium Press
8285 S.W. Nimbus
Suite 151
Beaverton, Oregon 97005-6401
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploring ATARI's Keyboard #1–#7</td>
<td>1</td>
</tr>
<tr>
<td>Component 1 Fun Page</td>
<td>10</td>
</tr>
<tr>
<td>Programming Your ATARI #1–#5</td>
<td>12</td>
</tr>
<tr>
<td>Programmer's Pastime #1–#4</td>
<td>18</td>
</tr>
<tr>
<td>Component 2 Fun Page</td>
<td>24</td>
</tr>
<tr>
<td>Programmer's Pastime #5–#18</td>
<td>26</td>
</tr>
<tr>
<td>Component 3 Fun Page</td>
<td>49</td>
</tr>
<tr>
<td>Programmer's Pastime #19–#28</td>
<td>51</td>
</tr>
<tr>
<td>Component 4 Fun Page</td>
<td>64</td>
</tr>
<tr>
<td>Programmer's Pastime #29</td>
<td>66</td>
</tr>
<tr>
<td>Component 5 Fun Page</td>
<td>89</td>
</tr>
<tr>
<td>Programmer's Pastime #38–#53</td>
<td>91</td>
</tr>
<tr>
<td>Component 6 Fun Page</td>
<td>134</td>
</tr>
<tr>
<td>Programmer's Pastime #54–#71</td>
<td>136</td>
</tr>
<tr>
<td>Component 7 Fun Page</td>
<td>178</td>
</tr>
</tbody>
</table>
KEYBOARD ILLUSTRATIONS

ATARI 800

ATARI XL
EXPLORING ATARI'S
KEYBOARD #1

Take a few minutes to explore ATARI's keyboard. Press single keys, and keys while SHIFT CTRL are being held down, and see what happens. (The keys on the far right row are for commercial programs. Do not press them.)

Finish drawing the key or keys that must be pressed to get ATARI to type what is shown on the screen at the right. Check your answers by using ATARI.

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 

A
33
#
v
<
;
1. Turn ATARI on.

2. Press \texttt{RETURN}.
   What did the cursor do? 

3. Press \texttt{SPACE} (It's the long, unlabeled bar at the bottom.)
   What did the cursor do this time? 

4. Press \texttt{SHIFT} and \texttt{CLEAR} together.
   What did the cursor do? 

5. Type your name (first, middle, and last).

6. Hold \texttt{CTRL} and press \texttt{~} five times.
   Which way did the cursor move? 
   Did anything happen to the writing on the screen? 

Now you know how to move the cursor around the screen without erasing any of the writing.

7. Hold \texttt{CTRL} and press the various cursor control keys (those with arrows \texttt{\leftarrow \rightarrow \uparrow \downarrow}).
   Notice how the cursor moves around the screen without changing the writing.

8. Use the \texttt{CTRL} and cursor control keys (with arrows) to move the cursor to the first letter of your middle name.
9. Press the space bar two times.
   What happened?

10. Press the DELETE key five times.
    What happened?

11. Hold SHIFT while pressing CLEAR.
    What happened?
FINISH DRAWING THE KEY (OR KEYS) THAT MUST BE
PRESSED TO GET ATARI TO PERFORM EACH SPECIAL
FUNCTION.

1. ___________ ___________ Move the cursor up.

2. ___________ ___________ Clear the screen and
send the cursor home.

3. ___________ ___________ Move the cursor right.

4. ___________ ___________ Begin reverse field
printing.

5. ___________ ___________ End reverse field
printing.

6. ___________ ___________ Move the cursor left.

7. ___________ ___________ Delete a letter.

OR ___________
8. Delete a line.

9. Insert a space.

10. Insert a line.
1. Turn ATARI on.
2. Type I LIKE YOU ATARI.
3. Press DELETE BACKS five times.
   What happened?

4. Hold SHIFT and press INSERT five times.
   What happened?

5. Hold SHIFT and press DELETE BACKS five times.
   What happened?

6. Hold SHIFT and press CLEAR.
   What happened?

7. Type I LOVE YOU ATARI.
8. Hold CTRL and press ← until the cursor is on the Y of YOU.
9. Hold CTRL and press DELETE BACKS four times.
   What happened?

    What happened?

11. Type YOU back in the new space.
12. If you have time, try typing some lines of your own, and use the various keys to do some screen editing.
Quick Review

Tell what happens when these keys are pressed:

1. SHIFT CLEAR <
   
2. SHIFT INSERT >
   
3. SHIFT DELETE BACK SPACE
   
4. CTRL INSERT >
   
5. CTRL DELETE BACK SPACE
   
6. CTRL ↑
   
7. CTRL ↓
   
8. CTRL ←
   
9. CTRL →
   
10. (SPACE)
   
11. DELETE BACK SPACE
EXPLORING ATARI'S
KEYBOARD #6

Mine the Diamonds
created by Wendy Cheldelin

1. Turn ATARI on.

2. Clear the screen.

3. Press \textbf{CTRL} and hold it down. Press the \textbf{P} key 5 times.

4. Press \textbf{CTRL} and hold it down. Now press the \textbf{1} key once. There should be 5 "rocks" and 1 "diamond" on your screen.

5. On the same line, make 4 or 5 more rocks followed by 1 diamond until the line is filled and the cursor has moved to the line below.

6. Fill one more line with rocks and diamonds.

7. Now the challenge begins! Mine the diamonds by erasing all of the rocks. Be careful! Don't erase any diamonds.
Design Your Own Game!

1. Design a game that requires some understanding of ATARI's keyboard. Write clear directions on how to play your game in the space below.

2. Ask a friend to play your game!
**down**

1. The language that ATARI speaks.
2. What type of computer is ATARI?
3. What is another word for "erase"?
4. We must always press this key when we are done typing a line.
5. Hold this key down as you press another key and ATARI will print the symbol at the top or front of the key.
6. The set of directions a computer uses.
7. Symbols used to make pictures and borders.
8. Something that both a computer and a typewriter have.
9. The part of ATARI that shows what is typed.
10. Always press this key between words and numbers that you type.
11. The space it takes to store one letter in ATARI's memory.
12. What is another word for "add"?
13. The white square that shows you where ATARI will type next on the screen.
14. Means "record".
15. The place where ATARI remembers programs.

Evaluate Yourself

1. In component 1, I did __________________________ because __________________________
2. Component 1 was __________________________
3. Tell about the good parts of the component:
   __________________________
4. Tell about the bad parts of the component:
   __________________________
Speak

1. Write a program that tells ATARI to print:

   COMPUTER PROGRAMMING IS FUN!

2. Make sure each line of your program begins with a line number.

3. Check your program to make sure there are no mistakes.

4. RUN your program on the computer.

Use this format

```
10 PRINT "" ""
20 END
```

Write your program here
Speaking Nonstop

1. Write a program that tells ATARI to print your name over and over again!
2. RUN your program on ATARI.

Use this format

10 PRINT ""  "")
   (Type your name inside the quotes.)
20 GOTO 10

Write your program here
Top-Secret

1. Your mission is to write a program that tells ATARI to print a top-secret message in a secret code. Use the graphic symbols on the keys as your code. For example, if we wanted a word in our message to say SAW, the code would be + r t because these graphic symbols appear on the S, A, and W keys.

2. Give your program to a friend. Have your friend RUN it on ATARI and try to decode the secret message.

Your success as a secret agent depends on this program. Good luck! (This page will self-destruct in two days if your program is not finished.)

Write your program here
Computer Art

1. Write a program that tells ATARI to print a design using graphic symbols.
2. Make it so that your design will be printed over and over on the screen.
3. RUN the program on ATARI.

Write your program here
Shuttle Launch

1. You can program ATARI to make moving pictures by using these five features:
   - line numbers
   - PRINT statements
   - quotation marks
   - GOTO statement
   - graphic symbols

2. Use the format below to create a program that launches a rocket.

3. RUN the program on ATARI.

Use this format

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>20</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>30</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>40</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>50</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>60</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>70</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>80</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>90</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>100</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>110</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>120</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>130</td>
<td>PRINT</td>
<td>&quot;</td>
</tr>
<tr>
<td>140</td>
<td>GOTO</td>
<td>10</td>
</tr>
</tbody>
</table>

NOTE: It is important to leave lines 120 and 130 blank inside the quotation marks so the rockets are spaced out when they move on the screen.
Write your program here

You may have more than 140 lines in your program.
Write the symbol that ATARI uses for each arithmetic operation.

addition _____ multiplication _____
powers _____ subtraction _____
division _____ square root _____

How would you type each equation to get answers from ATARI?
1. \(457 + 99 \times 6\)  
2. \(\sqrt{64}\)  
3. \(26 \div 2^2\)  
4. \(777 \times 555 \div 222\)  
5. \(8^3 - 16\)  
6. \(\sqrt{22} + 88\)  
7. \(\sqrt{49} + 765\)  
8. \(98 + 88 \times 66 \div 2^4\)

Show how you would type the equations above using only one PRINT (?) statement.

__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________
__________________________________________________________________________


What order does ATARI follow in doing arithmetic for equations with many numbers,

_________________________ are done first.

_________________________ are done second.

_________________________ and ____________________
are done third (left to right).

_________________________ and ____________________
are done last (left to right).

Use your mental powers and write the answers that ATARI would give for these equations. Remember to do the arithmetic in the same order that ATARI would!

1. 2*3+1 ____________________
2. 2+8*2*1 ____________________
3. 3*3+9+20 ____________________
4. 11+4*3 ____________________
5. 22+8+12*1 ____________________

Try some more.

1. 8/2−3 ____________________
2. 20/4+6−5 ____________________
3. 30−10/2 ____________________
4. 50−20/10+3 ____________________
5. 16/2+8/2 ____________________
6. 14/2+4/2−6 ____________________
Use your mental powers and write the answer ATARI would give for these equations.

1. $4*4+6/2$
2. $8/4+3*3$
3. $20-(4+5*2)+15$
4. $6+14/7*5$
5. $(9/3-2)*(7*2+4)$
6. $(7+2-4+6*1)/1$

**POWER!**

Powers are also called EXPONENTS. Read the following examples and figure out how powers work.

1. $10^1=10*1=10$
2. $10^2=10*10=100$
3. $10^3=10*10*10=1000$

Try some more:

1. $2^1=2*1=2$
2. $2^2=2*2=4$
3. $2^3=2*2*2=8$
4. $2^4=2*2*2*2=16$

Identify the powers by filling in the blanks:

1. $3^1=$
2. $3^2=$
3. $3^3=$
4. $3^4=$

Now try these:

1. $4^1=$
2. $4^2=$
3. $4^3=$
4. $4^4=$
CHALLENGE

1. $2 \times 4 \times 5$
2. $5 \times 2 \times 4$
3. $7 + 3 \times 2$
4. $5 \times 2 - 13$
5. $100 / 10 \times 2$
6. $12 \times 2 - 5 \times 10$
7. $8 \times 2 \times (4 + 5)$
8. $200 - (6 + 3 \times 3) + 7$

(REMEMBER—If you check these on the computer, ATARI gives a slightly inaccurate answer when working with powers.)

In each of the following equations, put a 1 under the part the computer would do first, a 2 under the second part it would do, and so on.

Example: $82 + 72 / 2 - (4 + 22) + 9 \times 3$

\[
\begin{array}{cccc}
 4 & 3 & 5 & 1 \\
\end{array}
\]

1. $3000 - (13 - 6 \times 4) + 8 \times 3 + 9$

\[
\begin{array}{cccc}
 4 & 3 & 5 & 1 \\
\end{array}
\]

2. $900 / 7 \times 2 + (16 - 9 \times 4) \times 3$

\[
\begin{array}{cccc}
 4 & 3 & 5 & 1 \\
\end{array}
\]
Cowboy Clyde typed in the following equations, but ATARI wouldn’t give him answers. Do you know why?

Find out what’s wrong with the way his equations are typed. Write the correct way in the blanks.

1. \(?62+4\times20\)  
2. \(23/4\times6\)  
3. \(?SQR\ 16\)  
4. \(80/4+2\times3\times\ SQR\ 25\)

**CHALLENGE**

If there’s some arithmetic in a long equation that you want to be done first, put parentheses around it. (ATARI always does what’s in parentheses first.) Let’s say you want \(4+3\times2\) to equal 14. If you type: \(4+3\times2\) ATARI will give you 10 because multiplication is done before addition. So it becomes \(4+6\) which equals 10.

If you want \(4+3\times2\) to equal 14, you must use parentheses like this: \((4+3)\times2\). First

\[7\times2=14\]
Rewrite each equation below and put parentheses around what ATARI should do first in order to make the equation TRUE. HINT: You might have to use two sets of parentheses for some equations.

**Example:** $7 \times 3 + 2 = 35$

1. $9/2 + 1 = 3$
2. $6 \times 2 + 2 = 24$
3. $3 \times 3 - 1 = 9$
4. $4 + 2 - 1 \times 5 = 9$
5. $12 - 3 + 6/3 = 9$
6. $20 - 10 \times 4 + 2 = 10002$
7. $12/4 + 2 = 2$
8. $9 - 5 \times 2 = 16$
9. $50 + 10/10 + 10 + 2 = 2$
10. $10 + 4 - 7 \times 2 + 1 + 3 - 3 = 193$

An * means it's an extra tough problem!
Down

1. In a long equation, ATARI does _______ second.
3. Type the ______ command when you want to see ATARI do a trick or run a program.
5. Which command do you type when you want to clear ATARI's memory or do something new?
7. Which statement in a program tells ATARI that your program has ended?
9. Which program statement tells ATARI to write something?

10. A ______ mark is a shortcut used in place of a PRINT statement.
11. Many equations on one line must be separated by ______.
12. Using ATARI as a calculator is called ______ mode programming.
13. In a program, 10
    20
    30 are called ______ numbers.

Across

1. In a long equation, ATARI does the operation in ______ first.
2. Which command do you type when ATARI is finished loading a game and ready to play?
4. Put ______ marks around what you want ATARI to say in a program.

6. Every step in a program must begin with a line ______.
8. Which key should you press when you are done typing a line and you want to go to the next line?

Evaluate Yourself

1. In component 2, I did _____________________________ because _____________________________.

2. Component 2 was _____________________________.

3. Tell about the best parts of the component:
   _____________________________

4. Tell about the parts of the component that were hard for you: _____________________________

5. Tell about the parts of the component that you like the least: _____________________________

25
For each flow chart, fill in the blank boxes with the step you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart #1
How to feed your pet elephant.

START

Call your pet elephant.

Put the peanuts in the bowl

Get out the peanuts.

STOP

Missing Steps
Algorithm/Flow Chart #2
How to wash your pet skunk.

**Missing Steps**

Get skunk wet

STOP

Rinse skunk

Fill tub with warm water

START

Dry skunk
For each flow chart, fill in the blank boxes with the step you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart #1
How to walk your pet alligator.

1. **Missing Steps**
2. Attach leash to collar
3. Go for a walk
4. START
5. Muzzle the alligator
6. STOP
7. Put on his collar

28
Algorithm/Flow Chart #2
How to make an ice cream cone.

**Missing Steps**

Put away ice cream

Scoop out ice cream

START

Get out ice cream

STOP

Get out a cone

Put scoop of ice cream into cone

Pack the ice cream firmly into cone

Eat ice cream cone

...
Design an algorithm for how to make a peanut butter and banana sandwich. Write your algorithm in flow chart form.
For each flow chart, fill in the blank boxes with the steps you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart #1:
How to take a hot bath.

Missing Steps
Fill tub with water
Enjoy your bath
START
Turn off hot water
Turn off water
STOP
Is water hot enough?
Plug drain in tub
Turn on more hot water
Algorithm/Flow Chart #2
How to eat a bowl of Rice Krispies.

This flow chart has two single-alternative decision steps!

Missing Steps
Eat your cereal
Get out bowl & spoon
STOP
Add milk
Pour Rice Krispies into bowl
Is there too much cereal in your bowl?
Get out Rice Krispies and milk
Remove some cereal from bowl and put back into box
START

Add more milk

Do you hear snap, crackle, and pop?

YES

NO
Here are the steps of an algorithm to make a chocolate milkshake. Put the steps in order and make a flow chart. Be sure to show the SINGLE-ALTERNATIVE DECISION STEP.

**Steps**

STOP

Add \( \frac{1}{8} \) cup chocolate sauce

Get out blender

Put 3 scoops of ice cream into blender

Is milkshake too thick?

Get out ingredients

Add 1 cup milk

Blend all ingredients

Add more milk and blend ingredients again

Pour milkshake into glass

START

Drink your milkshake
Write an algorithm on how to fix yourself a cold glass of water. Make the algorithm into a flow chart. Your flow chart should have a SINGLE-ALTERNATIVE DECISION STEP.

Ask the question, "IS THE WATER COLD ENOUGH?"

For the NO answer detour, your step might say:
ADD ICE CUBES.
For each flow chart, fill in the blank boxes with the steps you think would fit. Make sure your steps are in the right order.

Algorithm/Flow Chart: How to teach your pet bull to come when you call.

**Missing Steps**

- Hold out handful of hay
- Climb out of corral
- START
- Gently get bull’s attention by calling his name
- Pet bull. Quickly give him the hay
- Say “COME TORO” over and over
- Turn and run as fast as you can
- STOP
- Climb into corral. Approach bull carefully
- Is the bull charging at you?
Algorithm/Flow Chart: How to bake cookies.

**Missing Steps**

Wait 10 min. then open oven

STOP

Are cookies done?

Gobble the cookies

Mix ingredients according to recipe

Cool 5 minutes

Put cookie sheets in hot oven

Remove cookies from oven

Put dough on cookie sheet

START

Wait 3 min. Then turn off oven

NO

DOUBLE-ALTERNATIVE DECISION STEP

YES

Turn Off oven

36
Write an algorithm and flow chart on how to hit a baseball with a bat during a game when you are up to bat. Your flow chart must have a DOUBLE- ALTERNATIVE DECISION STEP.

Question: IS THE BALL IN THE STRIKE ZONE?

**YES**
KEEP YOUR EYE ON THE BALL

**NO**
STEP AWAY FROM THE PLATE
Fill in the blank boxes of the flow chart with the step you think should fit. Make sure the steps fit the loop.

Algorithm/Flow Chart: How to eat candy.

**Missing Steps**

- Is there more candy left?
- Brush your teeth
- STOP
- Throw away wrapper
- Unwrap candy
- START
- Chew and swallow
- Take a bite
Write an algorithm for how to wash your hair. Write the algorithm in flow chart form. Your flow chart should have a LOOP.

**Example**

**Question:** IS THERE SOAP IN YOUR HAIR?
**LOOP:** YES → RINSE YOUR HAIR
For each algorithm/flow chart, write a program in BASIC. (HINT — Pressing CTRL and ⌥ will give a tree-like graphic.)

1. START

   ↓

   Print
   "SAVE OUR TREES"

   ↓

   PRINT
   "♠ ♠ ♠"

2. START

   ↓

   PRINT
   "My favorite friend is"

   ↓

   PRINT
   "You!"
3. START
   ↓
   Print "4*400 ="
   ↓
   Print 4*400
   ↓
   STOP

4. START
   ↓
   Print "10^2 ="
   ↓
   Print 10^2
   ↓
   Print "10^3 ="
   ↓
   Print 10^3
   ↓
   STOP
Write an algorithm in a flow chart for each problem. Then write a BASIC program for ATARI to follow.

1. Tell ATARI to print PETER PIPER over and over.
PICKED A PECK
OF PICKLED PEPPERS

Flow chart Program

2. Tell ATARI to print $100 \times 10 - 1000 = 0$

Flow chart Program
3. Tell ATARI to print

\[
\begin{array}{c}
\text{I CAN DO} \\
56789 \times 1234 \\
\text{WHICH EQUALS} \\
70077626
\end{array}
\]

Flow chart       Program
4. Tell ATARI to print

SOMEWHERE
OVER THE RAINBOW
MANY COMPUTERS TRY
EQUATIONS LIKE
4*10=
(answer to 4*10)

**Flow chart**

**Program**
For each program, write what you think ATARI would print.

**Example:**

10 ? "10+500+200=
20 ? 10+500+200
30 GOTO 20

<table>
<thead>
<tr>
<th>ATARI would print</th>
</tr>
</thead>
<tbody>
<tr>
<td>10+500+200=</td>
</tr>
<tr>
<td>710</td>
</tr>
<tr>
<td>710</td>
</tr>
<tr>
<td>(The dots mean that</td>
</tr>
<tr>
<td>710 would be</td>
</tr>
<tr>
<td>printed forever.)</td>
</tr>
</tbody>
</table>

1. 10 ? "600-400+64="
20 ? 600-400+64
30 END

<table>
<thead>
<tr>
<th>ATARI would print</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

2. 10 ? "I LOVE "
20 ? "YOU !!! "
30 GOTO 20

3. 10 ? "SHE SELLS"
20 ? "SEA SHELLS"
30 ? "BY THE SEASHORE"
40 ? "I CAN SAY IT!"
50 GOTO 40
For each program, show how ATARI would print the equation on the screen.

**Example:**
```
10 ? "10+37=", 10+37
20 END
```

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>+</td>
<td>3</td>
<td>7</td>
<td>=</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

(1st Print Zone) (2nd Print Zone)

1. 10 ? "66+33=", 66+33
```
20 END
```

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

(1st Print Zone) (2nd Print Zone)

2. 10 ? "88-44+2=", 88-44+2
```
20 END
```

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

(1st Print Zone) (2nd Print Zone)

3. 10 ? "100+200+300=", 100+200+300
```
20 END
```

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

(1st Print Zone) (2nd Print Zone)

---

At this rate I'll be tying my shoes forever.
4. 10 ? "10*50=\(\_\)";
   10*50
   20 ?
   30 ? "10*50=\(\_\)";
   10*50
   40 END

5. 10 ? "60/20=\(\_\)",
   60/20
   20 ?
   30 ? "60/20=\(\_\)";
   60/20
   40 END

6. 10 ? "5-6=\(\_\)"
   5-6
   20 END

7. 10 ? "8-7=\(\_\)"
   8-7
   20 ?
   30 ? "7-8=\(\_\)"
   7-8
   40 END

I'm done!!
Write a program that tells ATARI to print what is seen on the screens below.

<table>
<thead>
<tr>
<th>Screens</th>
<th>Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>8. ATARI IS A . . . NUMBER CRUNCHER</td>
<td></td>
</tr>
<tr>
<td>9. 10−9=</td>
<td>1</td>
</tr>
<tr>
<td>9−10= −1</td>
<td></td>
</tr>
<tr>
<td>10. 5−6= −1</td>
<td></td>
</tr>
<tr>
<td>6−5= 1</td>
<td></td>
</tr>
<tr>
<td>11. IF YOU CAN DO</td>
<td>8∧8</td>
</tr>
<tr>
<td>YOU’RE A . . .</td>
<td></td>
</tr>
<tr>
<td>WHIZ KID</td>
<td></td>
</tr>
<tr>
<td>WHIZ KID</td>
<td></td>
</tr>
<tr>
<td>WHIZ KID</td>
<td></td>
</tr>
</tbody>
</table>
Down

1. In the flow-diagramming process, we write our algorithm in a flow _______.
3. In a flow chart, one detour from a decision box is a ______-alternative decision step.
5. In a flow chart, a step that is repeated is called a _______.
7. The rectangle-shaped boxes in a flow chart that tell you to do something are called _______ boxes.

Across

2. An _______ is a step-by-step method that can be used to solve a problem.
4. Two detours from a decision box in a flow chart is a ______-alternative decision step.
6. Each print zone on ATARI's screen can hold _______ characters.
8. The BASIC command to loop is _______.

9. ATARI's screen has four print _______.
13. The last step in a flow chart is _______.
14. A _______ tells ATARI to go to the next print zone and then begin printing.
10. The diamond-shaped boxes in a flow chart that ask you questions are called _______ boxes.
11. The first step in a flow chart is _______.
12. A _______ tells ATARI to hold the cursor in place until the next PRINT command.
15. Each PRINT statement tells ATARI to print on a _______ line.

Evaluate Yourself

1. Component 3 was ________________________________ because ________________________________

2. The best parts of the components were ________________________________

3. The parts I liked the least were ________________________________

4. The most valuable thing I learned in this component was ________________________________
   because ________________________________
For each LET statement, fill in the CONTENTS of the memory cell mailbox.

1. 10 LET OP=33
2. 20 LET T2=17
3. 30 LET M1=3000
4. 40 LET D=640

For each LET statement, fill in the ADDRESS and the CONTENTS of the memory cell mailbox.

1. 10 LET FF=99
2. 20 LET PQ=5
3. 30 LET N=0
4. 40 LET W7=62

Read each variable below. If it follows the rules we learned for writing variables, write "YES." If it does not follow the rules, write "NO."

1. PR
2. ITQ
3. A
4. E1
5. 3X
6. 7Z
7. X3
8. BB
9. 2CC
10. 23D
11. FDd
12. KK1
Read each program. Then write what ATARI would print as the output. If you can, check your answers by running the programs on ATARI.

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 10 LET ZB = 14&lt;br&gt;20 ? ZB&lt;br&gt;30 END</td>
<td>ZB</td>
</tr>
<tr>
<td>2. 10 LET T2 = 77&lt;br&gt;20 ? &quot;T2&quot;&lt;br&gt;30 END</td>
<td>T2</td>
</tr>
<tr>
<td>3. 10 LET U = 182&lt;br&gt;20 ? &quot;U&quot;&lt;br&gt;30 ? U&lt;br&gt;40 END</td>
<td>U</td>
</tr>
<tr>
<td>4. 10 LET RC = 7&lt;br&gt;20 ? &quot;RC IS&quot;&lt;br&gt;30 ? RC&lt;br&gt;40 END</td>
<td>RC</td>
</tr>
<tr>
<td>5. 10 LET GG4 = 66&lt;br&gt;20 ? &quot;GG4 IS »&quot;&lt;br&gt;30 ? GG4&lt;br&gt;40 END</td>
<td>GG4</td>
</tr>
</tbody>
</table>
Read each program. Then write what ATARI would print as the output. If you can, check your answers by running the programs on ATARI. Pay close attention to safe and unsafe variables!

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| 1. 10 LET RB4 = 40  
   20 LET RB5 = 50  
   30 LET RB1 = 10  
   40 ? RB5, "IS\n";  
   50 ? RB1, "MORE\n";  
   60 ? "THAN\n"; RB4  
   70 END |

| 2. 10 LET T = 5  
   20 LET V = 25  
   30 ? "THE SQUARE  
   ROOT OF\n";  
   40 ? V, "IS\n"; T  
   50 END |

| 3. 10 ? "MY FAVORITE  
   NUMBER IS\n";  
   20 LET D = 333  
   30 ? D  
   40 GOTO 30 |

| 4. 10 ? "MY FAVORITE  
   NUMBER IS\n";  
   20 LET D = 333  
   30 ? D  
   40 GOTO 30 |
Read each program. Then write what ATARI would print as the output.

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| 1. 10 LET N=12  
  20 LET R=6  
  30 ? N/R  
  40 END | |
| 2. 10 LET N=12  
  20 LET R=6  
  30 ?"N+R=\$"; N+R  
  40 END | |
| 3. 10 LET N=12  
  20 LET R=6  
  30 ?N,"+";R,"="\$";  
  N+R  
  40 END | |
| 4. 10 LET Q=10  
  20 LET R=20  
  30 LET S=30  
  40 LET T=40  
  50 ?T/R  
  60 ?Q,"+";S,"="\$";  
  T  
  70 ?S-Q,"="\$";S;  
  "-";Q  
  80 END | |
For each LET statement, fill in the contents of the memory cell mailboxes.

1. 10 LET P = 41
    20 LET Q = P

2. 10 LET A = 36
    20 LET E = 16
    30 LET I = A

3. 10 LET L = 4
    20 LET M = 2
    30 LET N = 4 + 2

4. 10 LET B1 = 3
    20 LET B2 = 6
    30 LET B3 = B1 + B2

5. 10 LET AB = 10
    20 LET AC = 15
    30 LET AD = AC + 5
    40 LET AE = AB + 10

6. 10 LET GP = 19
    20 LET GQ = GP - 4
    30 LET GR = GQ + 4
    40 LET GS = GR * 1
Read each program. Then write what ATARI would print as the output. Check your answers by running the programs on ATARI.

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| 1. 10 LET PJ=17  
  20 LET J2=34  
  30 LET J4=PJ+J2  
  40 ? J4  
  50 END |
| 2. 10 LET B=2  
  20 ? B  
  30 LET B=100  
  40 ? B  
  50 END |
| 3. 10 LET X1=2  
  20 LET X2=X1*5  
  30 LET X3=X2/X1  
  40 ? X3  
  50 END |
| 4. 10 LET E6=3  
  20 LET E7=12  
  30 ? "PRODUCT", 'QUOTIENT'  
  40 ? E6*E7, E7/E6  
  50 END |
Program

5.10 LET HI=16
  20 LET HJ=HI+4
  30 ?HJ+10
  40 END

6.10 LET M=16
  20 LET N=14
  30 ?M+N
  40 LET N=12
  50 ?M+N
  60 END

7.10 LET Z1=8
  20 LET Z2=Z1-2
  30 ?Z2+Z1/2
  40 END

8.10 LET T1=6
  20 LET T1=7
  30 ?T1
  40 GOTO 30
  50 END

9.10 LET J=11
  20 LET K=22
  30 LET J=17
  40 ?K+J
  50 END

Output
Read each program. Then write what ATARI would print as the output. Check your answers by running the programs on ATARI.

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| 1. 10 LET A = 100  
20 LET B = A / 25  
30 LET C = B * A  
40 ? "B =" ; B  
50 ? "C =" ; C  
60 ? "IF A + B + C =" ;  
   A + B + C  
70 ? "THEN A =" ; A  
80 END |
| 2. 10 LET Q1 = 8  
20 LET Q2 = 4  
30 ? Q1 / Q2  
40 LET Q1 = 12  
50 ? Q1 / Q2  
60 END |
| 3. 10 LET C = 9  
20 LET D = 8  
30 LET C = 7  
40 ? C  
50 ? D  
60 END |
In each program there are one or more mistakes. Find the mistake(s) and circle the line number where you found it. Then write the statement the correct way in the space to the right.

**Program**

Example:

10 LET 4=D
20 ?D
30 END

**Correction**

10 LET D=4

1. 10 LET L=44
   20 LET 6=M
   30 ?L+D
   40 END

2. 10 ?Y
   20 LET Y=66
   30 END

3. 10 LET 5B=6
   20 LET H=3
   20 ?H
   40 END

4. 10 ?"A+B"; A+B
   20 LET A=3
   30 LET B=4
   40 END
Program                  Correction
5. 10 LET XY=5
    20 LET 3=VW
    30 ?XY+VW
    40 END

6. 10 LET X=99
    20 LET C=4
    ?X-C
    40 END

7. 10 LET D+2=10
    20 LET E=4
    30 ?D-E
    40 END

8. 10 LET 3Y=2
    20 LET B1=9
    30 ?B1
    40 END
Using any shortcuts you have learned so far, rewrite the long programs below to make them as short as possible.

<table>
<thead>
<tr>
<th>Program</th>
<th>Shortcut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>10 LET B=49</td>
<td>10 LET B=49; LET E=409</td>
</tr>
<tr>
<td>20 LET E=409</td>
<td>20 ?B+E</td>
</tr>
<tr>
<td>30 PRINT B+E</td>
<td>30 END</td>
</tr>
<tr>
<td>40 END</td>
<td></td>
</tr>
<tr>
<td>1. 10 LET R=50</td>
<td></td>
</tr>
<tr>
<td>20 LET N=22</td>
<td></td>
</tr>
<tr>
<td>30 ?R-N</td>
<td></td>
</tr>
<tr>
<td>40 ?R+N</td>
<td></td>
</tr>
<tr>
<td>50 END</td>
<td></td>
</tr>
<tr>
<td>2. 10 LET S1=98</td>
<td></td>
</tr>
<tr>
<td>20 LET S2=89</td>
<td></td>
</tr>
<tr>
<td>30 LET S3=889</td>
<td></td>
</tr>
<tr>
<td>40 PRINT &quot;S3-S2-S1=&quot;;</td>
<td></td>
</tr>
<tr>
<td>S3-S2-S1</td>
<td></td>
</tr>
<tr>
<td>50 PRINT &quot;S2*S1=&quot;;</td>
<td></td>
</tr>
<tr>
<td>S2*S1</td>
<td></td>
</tr>
<tr>
<td>60 END</td>
<td></td>
</tr>
<tr>
<td>3. 10 LET U=40</td>
<td></td>
</tr>
<tr>
<td>20 LET T=20</td>
<td></td>
</tr>
<tr>
<td>30 ?U+T</td>
<td></td>
</tr>
<tr>
<td>40 ?U-T</td>
<td></td>
</tr>
<tr>
<td>50 ?U*T</td>
<td></td>
</tr>
<tr>
<td>60 ?U/T</td>
<td></td>
</tr>
<tr>
<td>70 END</td>
<td></td>
</tr>
</tbody>
</table>
Program

4. 10 LET Y2=2
    20 LET Y3=22
    30 LET Y4=Y2+2
    40 LET Y5=Y3+22
    50 PRINT "Y4=", Y4
    60 PRINT "Y5=", Y5
    70 END

5. 10 LET D2=9
    20 LET D4=18
    30 ? "D2+D4=";
    40 ? D2+D4
    50 ? "D4−D2=";
    60 ? D4−D2
    70 END

Shortcut
For each E notation number, write the whole number or decimal that it stands for.

<table>
<thead>
<tr>
<th>E Notation</th>
<th>Whole number or decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example</strong>: 7.982E+07</td>
<td>79820000</td>
</tr>
</tbody>
</table>

1. 3.26E+11
2. 1.23E-04
3. 7.2E+08
4. 4.679E-06
5. 2.22E+07
6. 6.46982E-03
7. 5.3211E+12
8. 9.011E-05
9. 8.0045E+13
10. 1.467E-07
11. 8.006E+09
12. 6.002E-03
13. 3.9826E-05
14. 1.976345E+08
**Down**

1. How many pieces of information can a memory cell store?

3. If ATARI sees a variable that has not been introduced by a LET statement, ATARI will automatically give that variable a value of _______.

5. ATARI's memory can be thought of as electronic _______.

7. Whatever ATARI prints is called _______.

9. The _______ statement assigns a value to a variable.

11. Information is also called _______.

13. In a LET statement, the variable must always come before the _______.

15. When you introduce the same variable more than once in a program, ATARI will remember the _______ value you gave it.

16. We can use commas and _______ to change the arrangement of the output in a program.

17. ATARI will change numbers with more than ten digits into E _______.

---

**Across**

2. Which part of ATARI's brain allows it to do many of the tricks we teach it?

4. A mailbox in ATARI's memory is also called a memory _______.

6. A memory cell consists of the address and the _______.

8. Each memory cell mailbox has its own _______.

10. The address of a memory cell is also called a _______.

12. We can use _______ to shorten our program in between LET statements.

14. The type of number ATARI can't understand is a _______.

---

**Evaluate Yourself**

1. Component 4 was ___________________________ because ___________________________.

2. The best parts of the component were ___________________________.

3. The parts I liked the least were ___________________________.

4. The most valuable thing I learned in this component was ___________________________ because ___________________________.

---

65
Read each program. Then write what you think ATRARI would print as the OUTPUT. Run the programs on ATARI to check your answers.

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| 1. 10 FOR Q=2 TO 6  
20 ? Q  
30 NEXT Q  
40 END |        |
| 2. 10 FOR Q=2 TO 4  
20 ? "Q=";Q  
30 NEXT Q  
40 END |        |
| 3. 10 FOR A=1 TO 5  
20 ? "HELLO FRIEND!"  
30 ? "HOW ARE YOU?"  
40 NEXT A  
50 END |        |
| 4. 10 FOR D=1 TO 3  
20 ? D  
30 ? D+10  
40 NEXT D  
50 END |        |
| 5. 10 LET P=3  
20 FOR Q=1 TO 3  
30 ? P;"+";Q;"="; P+Q  
40 NEXT Q  
50 END |        |
Program

6. 10 FOR B=1 TO 5
   20 ?"B", "B+B", "B*B"
   30 ? B, B+B, B*B
   40 NEXT B
   50 END

7. 10 ? "MULTIPLICATION"
    TABLE FOR 7"
    20 FOR K=1 TO 12
    30 ? K, "TIMES 7 = ", K*7
    40 NEXT K
    50 END

8. 10 FOR G=1 TO 10
    20 ? "◊"
    30 NEXT G
    40 END

9. 10 FOR G=1 TO 10
    20 ? "◊";
    30 NEXT G
    40 END

10. 10 FOR S=1 TO 10
     20 LET S=S*S
     30 ? S, S/S
     40 NEXT S
     50 END

Can you explain how this program works? _

_
Write a program for each flow chart. Be sure to use a FOR–NEXT loop. Run your programs on ATARI to make sure they work.

**Flow chart**                        **Program**

1.  
   ![Flow chart 1](image1.png)

   START
   
   ↓
   
   FOR X = 1 TO 10
   
   ↓
   
   PRINT "I'M A PROGRAMMER"
   
   ↓
   
   NEXT X
   
   ↓
   
   STOP

2.  
   ![Flow chart 2](image2.png)

   START
   
   ↓
   
   FOR Y = 1 TO 50
   
   ↓
   
   PRINT Y
   
   ↓
   
   NEXT Y
   
   ↓
   
   STOP
3. START

PRINT "F", "F − F", "F/F"

FOR F = 1 TO 10

PRINT F, F − F, F/F

NEXT F

STOP
Flow chart

4. START

↓
PRINT
"MULTIPLICATION TABLE FOR 8"

↓
FOR J = 1 TO 12

↓
PRINT J*8

↓
NEXT J

↓
STOP
For each program description, write an algorithm in flow chart form. Then write the program. Run each program on ATARI to make sure it works.

**Description**  
5. Add something new to the program in #4 so it prints: `J; "TIMES 8="; J*8` each time the loop is done.

6. Write a program that prints the numbers from 1 to 20, their squares (X*X), and their cubes (X*X*X).
<table>
<thead>
<tr>
<th>Description</th>
<th>Flow chart</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>7. Write a program that prints (*) 20 times on one line.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8. Write a program that introduces \(D = 5\) and \(P = 1\) to 5. Make the program add \(D\) plus each value of \(P\), and print the sums of \(D + P\). |
In each program there is one mistake. Find the mistake and circle the line number where you found it. Then write the statement the correct way in the space to the right.

<table>
<thead>
<tr>
<th>Program</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 10  FOR P = 1 - 40</td>
<td></td>
</tr>
<tr>
<td>20 ?P</td>
<td></td>
</tr>
<tr>
<td>30 NEXT P</td>
<td></td>
</tr>
<tr>
<td>40 END</td>
<td></td>
</tr>
</tbody>
</table>

| 2. 10  FOR W IS 6 TO 30 |
| 20 ?W |
| 30 NEXT W |
| 40 END |

| 3. 10  FOR E = 3 TO 10 |
| 20 ?E |
| 30 E NEXT |
| 40 END |

| 4. 10 FOR L = 1 TO 4 |
| 20 ?L |
| 30 ?L*2 |
| 40 |
| 50 END |

| 5. 10 LET G = 4 |
| 20 FOR H = 5 TO 9 |
| 30 ?G + H |
| 40 NEXT G |
| 50 END |
Read each program. Write what you think ATARI would print as the OUTPUT. Run the programs on ATARI to check your answers.

**Program**  
1. 10 FOR F=0 TO 8  
   STEP 2  
   20 ? F  
   30 NEXT F  
   40 END  

2. 10 FOR J=18 TO 0  
   STEP -3  
   20 ? J  
   30 NEXT J  
   40 END  

3. 10 FOR B2=3 TO 21  
   STEP 3  
   20 ? "HOWDY"  
   30 NEXT B2  
   40 END  

4. 10 LET N=5  
   20 FOR T=1 TO N  
   30 ? T  
   40 NEXT T  
   50 END  

5. 10 LET M2=10  
   20 FOR S=0 TO 12  
   STEP M2/5  
   30 ? S  
   40 NEXT S  
   50 END
Program
6.10 ?"IF SEPT = 1 IS
   A SUNDAY THEN"
20 FOR JJ = 1 TO 31
   STEP 7
30 ?"SEPT=": JJ;
   "IS A SUNDAY"
40 NEXT JJ
50 END

7.10 LET PX = 8
20 FOR A7 = 0 TO 10
   STEP PX/4
30 ? A7
40 NEXT A7
50 END

8.10 FOR ZZ = 1 TO 14
   STEP 4
20 ? ZZ
30 NEXT ZZ
40 END

9.10 FOR BD = 20 TO 2
   STEP -5
20 ? BD
30 NEXT BD
40 END
For each program description, write an algorithm in flow chart form. Then write the program. Run each program on ATARI to make sure it works.

<table>
<thead>
<tr>
<th>Description</th>
<th>Flow chart</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Write a program that tells ATARI to count from 0 to 40 by fours.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Write a program that tells ATARI to count backwards from 8 to 0.
<table>
<thead>
<tr>
<th>Description</th>
<th>Flow chart</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Write a program that tells ATARI to print &quot;HELLO&quot; five times. Use a STEP statement.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Write a program that tells ATARI to print the numbers 0 through 21 STEP N/4. Make N=12.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. Write a program that tells ATARI to print the numbers 1, 4, 7, 10 and 13. Use a STEP statement.
Study each program. Write what you think ATARI would print as the OUTPUT. Run each program to check your answers.

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| 1. 10 LET C=0  
   20 FOR FL=1 TO 100  
   STEP 10  
   30 ?"THINK"  
   40 LET C=C+1  
   50 ?C  
   60 NEXT FL  
   70 END |
| 2. 10 LET C=0  
   20 ?"BRAIN POWER"  
   30 LET C=C+1  
   40 ?C  
   50 GOTO 20 |
| 3. 10 LET C=0  
   20 FOR FL=1 TO 8  
   30 LET C=C+1  
   40 ?C  
   50 ?"AWESOME"  
   60 NEXT FL  
   70 END |
Program

4. 10 LET C=0
   20 FOR Z=1 TO 50 STEP 10
   30 ? "RAINBOW"
   40 LET C=C+1
   50 NEXT Z
   60 ? "I PRINTED"
   70 ? "RAINBOW"
   80 ? C; "TIMES"
   90 END

5. 10 LET C=0
   20 FOR FL=1 TO 4
   30 LET C=C+1
   40 ? C
   50 ? "JELLY BEANS"
   60 NEXT FL
   70 ? "A TOTAL OF"; C; " JELLY BEANS"
   80 END
Write a program for each flow chart, then run the programs.

Flow Chart 1.
1. START
   \rightarrow CLEAR SCREEN
   \rightarrow PRINT "THIS IS FUN!"
   \rightarrow FOR–NEXT TIME LOOP
   \rightarrow CLEAR SCREEN
   \rightarrow STOP

Flow Chart 2.
2. START
   \rightarrow CLEAR SCREEN
   \rightarrow FOR N = 1 TO 10
   \rightarrow PRINT N
   \rightarrow FOR–NEXT TIME LOOP
   \rightarrow CLEAR SCREEN
   \rightarrow NEXT N
   \rightarrow STOP
3. START
   ↓
   CLEAR SCREEN
   ↓
   PRINT "LOVE IS"
   ↓
   FOR-NEXT TIME LOOP
   → CLEAR SCREEN
   → PRINT "♡♡♡"

4. START
   ↓
   CLEAR SCREEN
   ↓
   FOR N = 1 TO 100
   ↓
   PRINT "COUNTING"
   ↓
   PRINT N
   ↓
   FOR-NEXT TIME LOOP
   ↓
   CLEAR SCREEN
   ↓
   NEXT N
   ↓
   CLEAR SCREEN
   ↓
   STOP
Flow Chart

5. START
   ↓
   CLEAR SCREEN
   ↓
   FOR N = 0 TO 50
     STEP 5
   ↓
   PRINT "COUNT BY FIVES"
   ↓
   SKIP A LINE
   ↓
   PRINT N
   ↓
   FOR-NEXT
     TIME LOOP
   ↓
   CLEAR SCREEN
   ↓
   NEXT N
   ↓
   CLEAR SCREEN
   ↓
   STOP

Program
You have learned how to program ATARI to move down a number of lines on the screen and then begin printing. To do this, you used a statement like this:

20 ?!??!?

This tells ATARI to move down 4 lines.

Now using the tricks you have learned to make ATARI clear the screen, and move the writing down several lines, write a program for the following flow charts. Try the programs out on ATARI.

**Flow Chart**

1. START
   - CLEAR SCREEN
   - MOVE 10 LINES DOWN
   - PRINT "SMART STUFF"
     - FOR T = 1 TO 1000: NEXT T
   - CLEAR SCREEN
   - STOP

**Program**
Flow Chart

2.

START

↓

CLEAR SCREEN

↓

PRINT "WHAT A GREAT TRICK!"

↓

MOVE DOWN 4 LINES

↓

FOR T = 1 TO 1000: NEXT T

Program
Write a program for each flow chart, then run your programs on ATARI to make sure they work.

Flow Chart

1. START
   ↓
   CLEAR SCREEN
   ↓
   FOR-NEXT TIME LOOP
   ↓
   PRINT "OFF AND ON"

This is the basic algorithm for making something blink.

2. START
   ↓
   CLEAR SCREEN
   ↓
   FOR-NEXT TIME LOOP
   ↓
   PRINT "WOWSERS"

At this rate I'll be tying my shoes forever.
Use your expertise and your imagination to write two of your own programs that make something blink. You can even make graphics or pictures blink! Don't be afraid to experiment.

**Flow Chart**

1.

**Program**
2.
Down

1. A FOR–NEXT loop creates _______ controlled loops in a program.
3. A counter also lets you keep track of how many times you have done a _______.
5. To slow down ATARI’s printing, use a FOR–NEXT _______ loop.
7. When you copy a program onto a cassette tape, you use the _______ command.
9. To see how a program is written, type _______.
11. We use the FOR–NEXT time loop to make things _______.

Across

2. A mistake in a program is called a _______.
4. Every FOR statement must have a _______ statement after it somewhere in the program.
6. We tell ATARI to count in a certain pattern by using the _______ command.
8. The statement ""ESC SHIFT CLEAR"" tells ATARI to _______ the screen.
10. We push _______ to make a program _______ running.
12. We _______ out of a program when we stop it before it has finished running.

Evaluate Yourself

1. Component 5 was ____________________________ because ____________________________

2. The best parts of the component were ____________________________

3. The parts I liked the least were ____________________________

4. The most valuable thing I learned in this component was ____________________________ because ____________________________

Other comments:
Study each program and write what you think ATARI would print as the output. Run the programs to check your answers.

<table>
<thead>
<tr>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
</table>
| 1. 5 DIM FS(10)  
10 LET FS = "44"  
20 LET F = 44  
30 ? FS, F  
40 END | |
| 2. 5 DIM GS(10)  
10 LET GS = "6+32="  
20 LET G = 6+32  
30 ? GS, G  
40 END | |
| 3. 5 DIM AS(10), SS(12)  
10 LET AS = "ADDITION"  
20 LET SS = "SUBTRACTION"  
30 ? AS, SS  
40 LET A = 8+8  
50 LET S = 8-8  
60 ? "8+8=", A, "8-8=", S  
70 END | |
| 4. 5 DIM QS(10), RS(10)  
10 LET QS = "HI HO"  
20 LET RS = "SILVER!"  
30 ? QS, RS  
40 GOTO 30 | |
| 5. 5 DIM ZS(20)  
10 LET ZS = "YOU'RE OUTA SIGHT!"  
20 FOR C = 1 TO 20  
30 ? ZS  
40 NEXT C  
50 END | |
There is at least one mistake in each program. Find the mistake(s), circle the line number where you found it, then write the statement(s) the correct way in the space to the right.

**Program**

1. 10 LET AZ$ = "YES"
    20 LET BY$ = "NO"
    30 ? AZ$, BY$
    40 END

2. 5 DIM T$(10), U$(10)
    10 LET T$ = "THE TIME"
    20 LET U$ = "IS NOW"
    30 ? T, U
    40 END

3. 5 DIM J$(10), K$(10)
    10 LET J$ = UP, UP
    20 LET K$ = AND AWAY
    30 ? J$, K$
    40 END

4. 5 DIM P$(20), T$(20)
    10 LET "PARTRIDGE IN" = P$
    20 LET "A PEAR TREE" = T$
    30 ? P$, T$
    40 END
Study each program. Write what you think ATARI would print as the output. You may write what you would answer for each INPUT statement. Run the programs to check your work.

Program

1. 5 DIM AS(20)
   10 ? "HOW OLD ARE YOU";
   20 INPUT A
   30 ? "WHAT'S IT LIKE TO BE\$"; A$
   40 INPUT A$
   50 ? "I'M GLAD TO HEAR IT'S\$"; A$
   60 END

2. 5 DIM AS(20)
   10 ? "HOW OLD ARE YOU?"
   20 INPUT A
   30 ? "WHAT'S IT LIKE TO BE\$"; A; ?
   40 INPUT A$
   50 ? "SO YOU ARE\$"; A$; "\$TODAY"
   60 END

3. 10 ? "HOW MANY BROTHERS AND"
   20 ? "SISTERS DO YOU HAVE?"
   30 ? "TYPE NUMBER OF BROTHERS,"
   40 ? "A COMMA,"
   50 ? "AND NUMBER OF SISTERS"
   60 INPUT B, S
   70 LET T=B+S
   80 ? "SO YOU HAVE\$"; T; "\$SIBLINGS"
   90 END
Program

4. 10 ? "CHOOSE TWO NUMBERS
    AND"
   20 ? "I WILL ADD THEM FOR YOU"
   30 ? "TYPE 1ST NUMBER, COMMA."
   40 ? "THEN TYPE THE 2ND NUMBER"
   50 INPUT F, S
   60 ?
   70 ? F,"+",S,"="; F+S
   80 ?
   90 ? "I'M A WHIZ!"
  100 END

5. 10 ? "TYPE IN 2 NUMBERS"
   20 ? "SEPARATED BY A COMMA"
   30 INPUT O, T
   40 ?
   60 ? O,"-",T,"="; O-T
   70 ? O,"*",T,"="; O*T
   80 ? O,"/",T,"="; O/T
   90 ? "SEE... I TOLD YOU"
  100 ? "I WAS A WHIZ!"
  110 END
Write a program for each flow chart. Run your programs on ATARI to check for bugs.

Flow Chart

1. START
   ↓
   DIMENSION VARIABLE
   ↓
   PRINT "WHAT'S YOUR NAME"
   ↓
   INPUT N$
   ↓
   PRINT "Hi$"; N$
   ↓
   STOP

I think I see a bug
Flow Chart

2.

START

↓

CLEAR SCREEN

↓

PRINT "TYPE 3 NUMBERS"

↓

PRINT "SEPARATED BY COMMAS"

↓

INPUT OE, TW, TH

↓

PRINT "OE;" + "TW; " + "TH;" = "; OE + TW + TH

↓

SKIP A LINE

↓

PRINT "THANK YOU!"

↓

STOP

Program
Flow Chart

3. START
   ↓
DIMENSION VARIABLE
   ↓
PRINT "WHAT IS YOUR ADDRESS?"
   ↓
INPUT A$
   ↓
SKIP A LINE
   ↓
PRINT "YOU LIVE AT: $A$
   ↓
STOP

Program
Write 3 programs using the INPUT statement. Write the flow chart for the algorithm first, then write the program. Debug your programs by running them on ATARI.

**Flow Chart**

**Program**
Write each equation as an IF-THEN statement.

**Question**

**Example:**
Is A equal to C?
1. Is L$ equal to "MAYBE"?
2. Is F1 not equal to F2?
3. Is GH greater than HI?
4. Is S$ less than or equal to F$?
5. Is X times B less than P times Q?
6. Is T divided by W greater than or equal to W times B?
7. Is P$ greater than M$?
8. Is the square root of Y equal to D?
9. Is G$ not equal to "NO"?
10. Is 10 divided by 5 less than 14 divided by 2?
11. Is Y$ equal to the square root of 64?
12. Is A plus B greater than D$?

**IF-THEN statement**

IF A = C THEN ____________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________

______________________________
For each question write the Complement IF-THEN statement.

**Example**
Is P$ equal to "YES"?

1. Is QR greater than 2?
2. Is Z$ not equal to "END"?
3. Is F less than or equal to P?
4. Is G$ equal to "JEEPERS"?
5. Is S1 greater than or equal to S2?
6. Is DD less than 444?
7. Is X greater than Y?
8. Is AS greater than or equal to 79?
9. Is P$ not equal to "YES"?
10. Is VP less than or equal to JK?
The location of the IF-THEN statement in a program is very important. If it is put in the wrong place, the program won't work properly. The IF-THEN statement must come after the LET or INPUT statements that introduce the variables in the IF-THEN statement. For example:

Program
10 IF P < Q THEN 50
20 LET P = 5
30 LET Q = 7
40 GOTO 60
50 ? "P IS SMALLER"
60 END

Output
ATARI does not print anything because the IF-THEN statement is before the LET statements that introduce the variables.

In one of the following programs, the IF-THEN statement is in the wrong place. Rewrite the programs so they are correct.

Incorrect program
1. 10 IF Z = 2 THEN 60
   20 LET A = 6
   30 LET B = 8
   40 LET Z = 2
   50 GOTO 70
   60 ? "Z = 2"
   70 END

Corrected program
Incorrect program

2. 5 DIM E$(10), D$(10)
10 ? "WHAT COLOR ARE YOUR EYES?"
20 IF ES = "BLUE" THEN 100
30 INPUT ES
40 ? "ARE THEY DIFFERENT COLORS?"
50 IF DS = "YES" THEN 120
60 INPUT DS
70 ? "THEY ARE 1 COLOR"
80 ? "THEY ARE NOT BLUE"
90 GOTO 130
100 ? "WHAT NICE BLUE EYES!"
110 GOTO 130
120 ? "WHAT COLORFUL EYES!"
130 END

Corrected program

3. 5 DIM FS$(10)
10 IF FS = "NO" THEN 60
20 ? "ARE COMPUTERS FUN?"
30 INPUT FS
40 ? "YOU'RE RIGHT!"
50 GOTO 70
60 ? "YOU'RE NO FUN!"
70 END
Study each flow chart, then write a program. Debug your programs by running them on ATARI.

Flow Chart

1. START
   ↓
   DIMENSION VARIABLE
   ↓
   PRINT "WHEN IS YOUR BIRTHDAY?"
   ↓
   INPUT B$
   ↓
   DOES B$ = "DECEMBER"?
   YES
   ↓
   NO
   PRINT "YOUR BIRTHDAY IS NOT IN DECEMBER"
   ↓
   GOTO STOP
   ↓
   PRINT "YOUR BIRTHDAY IS IN THE LAST MONTH OF THE YEAR"
   ↓
   STOP
Flow Chart          Program

2. START

↓
CLEAR SCREEN

↓
DIMENSION VARIABLES

↓
PRINT "HELLO"
"WHAT IS YOUR NAME?"

↓
INPUT N$

↓
SKIP A LINE

↓
PRINT "HOW ARE YOU TODAY?"

↓
INPUT H$

↓
SKIP A LINE

↓
PRINT "SINCE YOU ARE";H$;
"TODAY";N$

↓
PRINT "WE MUST GET TO WORK"

↓
STOP
Flow Chart

3. START
   ↓
   DIMENSION VARIABLES
   ↓
   PRINT "HELLO WHAT'S YOUR NAME?"
   ↓
   INPUT N$
   ↓
   SKIP A LINE
   ↓
   PRINT "HOW ARE YOU TODAY? FINE OR ROTTEN"
   ↓
   INPUT H$
   ↓
   DOES H$ = "ROTTEN"?
   ↓ NO
   ↓ YES
   PRINT "I'M SORRY" N$
   "THAT YOU ARE" H$
   ↓
   GOTO STOP
   ↓
   PRINT "GREAT";
   N$; "LET'S GET DOWN TO WORK"
   ↓ STOP

Program

Clue: You will need to use the complement of the question for your IF-THEN statement.
Flow Chart

4.

START

\[\text{DIMENSION VARIABLES}\]

\[\text{PRINT "WHAT IS YOUR NAME"}\]

\[\text{INPUT N$}\]

\[\text{SKIP A LINE}\]

\[\text{PRINT "HI"; N$; "DO YOU LIKE SCHOOL?"}\]

\[\text{INPUT S$}\]

\[\text{SKIP A LINE}\]

\[\text{DOES S$ = "YES" ?}\]

YES

\[\text{PRINT "GOOD FOR YOU! SEE YA"; N$}\]

\[\text{GOTO STOP}\]

NO

\[\text{PRINT "AW C'MON—BE A SPORT"; N$}\]

\[\text{STOP}\]

Program

Clue: You will need to use the complement of the question for your IF-THEN statement.
For each description, write an algorithm in flow chart form and write a program for the flow chart. Debug each program by running it on ATARI.

<table>
<thead>
<tr>
<th>Description</th>
<th>Flow Chart</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Alphabetize &quot;HIP&quot; and &quot;HIPPO&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Alphabetize "GUSTO" and "GROOVY"
3. Alphabetize
   "AARDVARK" and
   "ZEBRA"
Study each flow chart and then write a program. Use REM statements where appropriate to show good programming style. Debug your programs by running them on ATARI.

**Flow Chart**

1. 
   - START
   - DIMENSION VARIABLE
     - LET Q$ = "WHOOPEE"
     - FOR C = 1 TO 10
       - PRINT C
       - PRINT Q$
     - NEXT C
       - PRINT "DONE"
   - STOP
Flow Chart

2

START

↓

CLEAR SCREEN

↓

DIMENSION VARIABLES

↓

PRINT "WHAT IS YOUR FAVORITE COLOR?"

↓

INPUT C$

↓

DOES C$ = "RED"?

↓

YES

GOTO STOP

↓

NO

PRINT "YOU DON'T LIKE RED? RED IS MY FAVORITE COLOR"

↓

PRINT "I LOVE RED TOO!"

↓

STOP
3. START
   ↓
   CLEAR SCREEN
   ↓
   PRINT "60 + 40 = ":
   ↓
   INPUT S
   ↓
   DOES S = 100 ?
   YES
   PRINT "NO. TRY AGAIN"
   ↓
   GOTO
   ↓
   PRINT "GOOD JOB!"
   ↓
   PRINT "78 - 14 = ":
   ↓
   INPUT D
   ↓
   DOES D = 64 ?
   NO
   PRINT "NO TRY AGAIN"
   ↓
   GOTO
   YES
   PRINT "SUPER"
   ↓
   STOP
Study each program and write what you think ATARI would print as the OUTPUT—including error messages. Check your answers by running the programs on ATARI.

**Program**                  **Output**

1.  5 DIM ZS(10), LS(10)  
    10 READ ZS, LS  
    20 ? ZS, LS  
    30 GOTO 10  
    40 DATA "YOU",  
              "ARE", "A",  
              "HOT-SHOT"  
    50 END  

2.  10 DATA 4,14,41,6,  
    16,61,3,13,31  
    20 READ Q, R, S  
    30 ? Q+R+S  
    40 GOTO 20  
    50 END  

3.  10 READ G, H  
    20 DATA 44,66,88,  
              22,110  
    30 ? G, H  
    40 GOTO 10  
    50 END  

4.  10 DATA 14,7,2,16,8,  
    2,-99,-99,-99  
    20 READ A, L, B  
    30 IF A = -99 THEN  
        60  
    40 ? A - L - B  
    50 GOTO 20  
    60 END
**Program**

5. 10 READ R1, R2
    20 IF R1 = -1 THEN
        60
        30 ? R$ * R2
        40 GOTO 10
        50 DATA 2, 2, 3, 3, 4,
        7 4, 5, 5, -1, -1
        60 END

6. 5 DIM
    DS(10), ES(10)
    10 FOR L = 1 TO 4
    20 READ DS, ES
    30 ? DS, ES
    40 NEXT L
    50 DATA "A", "E",
        "I", "O"
    60 DATA "U", "Y",
        "ARE",
        "VOWELS"
    70 END

7. 10 FOR L = 1 TO 2
    20 READ S1, S2, S3
    30 DATA 8, 2, 4, 6, 2, 3
    40 ? S1 * S2 * S3
    50 NEXT L
    60 END
In each of the following programs there are mistakes. Circle the line number(s) with the mistake and make your correction in the space to the right. If something has been left out, add it to the program.

**Program** | **Correction**
--- | ---
1. 10 READ P, A, N  
20 ? P, A, N  
30 DATA 400, 8%, 6  
40 END  

2. 5 DIM NS(10)  
10 READ NS, A  
20 ? NS, A  
30 DATA KIM IS,  
3*4  
40 END  

3. 5 DIM NS(10)  
10 ? "NAME", "AGE"  
20 READ A, NS  
30 ? A, NS  
40 DATA HARVEY,  
14  
50 END  

4. 5 DIM NS(10)  
10 ? "NAME", "AGE"  
20 READ NS, A  
30 ? NS, A  
40 DATA HARVEY,  
14 YEARS OLD  
50 END
5.  5 DIM FS(20)
    10 READ FS
    20 ? "DAILY MENU"
    30 ? FS
    40 END

6.  10 READ X, Y, Z
    20 ? "THE PRODUCT
        OF 3 NUMBERS"
    30 ? X*Y*Z
    40 DATA 4.5
    50 END

7.  10 ? "COUNTING"
    20 READ DATA
    30 DATA 1,2,3,4
    40 END

8.  5 DIM NS(10)
    10 ? "NAME", "AGE"
    20 READ NS, A
    30 ? NS, A
    40 GOTO 20
    50 DATA BOB,
        BILL, 10, 11

---

Wanna drag?
READ-DATA statements can help you write shorter programs. Rewrite each program using READ-DATA statements to shorten them. Try to write each program so you don't get an error message.

**Long program**

1. 10 ? "MULTIPLYING 2 NUMBERS"
   20 LET P = 60
   30 LET Q = 129
   40 LET R = 410
   50 LET S = .6
   60 ? P, Q, P * Q
   70 ? R, S, R * S
   80 END

2. 5 DIM A$(10), B$(10), C$(10), D$(10), E$(10)
   10 ? "TEST SCORES"
   20 ? "NAME", "SCORE"
   30 LET A$ = "JOE"
   40 LET A = 98
   50 LET B$ = "TOM"
   60 LET B = 52
   70 LET C$ = "KRIS"
   80 LET C = 95
   90 LET D$ = "GAIL"
   100 LET D = 75
   110 LET E$ = "BOB"
   120 LET E = 72
   130 ? A$, A
   140 ? B$, B
   150 ? C$, C
   160 ? D$, D
   170 ? E$, E
   180 END
Flow Chart

4. START

↓

PRINT "FINDING
THE AREA OF
A TRIANGLE"

↓

FOR L = 1 TO 3

↓

READ
B,H

↓

PRINT B, H,
½*B*H

→ NEXT L

→ STOP

Program
2. START

PRINT "FINDING PERIMETER"

FOR X = 1 TO 3

READ S1, S2, S3, S4

PRINT S1, S2, S3, S4
 S1 + S2 + S3 + S4

NEXT X

STOP

3. START

PRINT "DIVIDING NUMBERS"

READ D1, D2

IS D1 = -9? YES

PRINT D1, D2, D1/D2

GOTO

STOP
2. Write a program that lists the names of your friends.

**Flow chart**

**Program**
Using what you know about READ-DATA statements:

1. Write a program that multiplies three numbers.

Flow chart

Program
4. Flow Chart

5. CODE the program

6. DEBUG

7. REVISE
Use the problem-solving approach to get ATARI to solve the following problems.

**Problem 1**
The teacher gave your class a test on programming the computer. The test scores were:

- Jill Jarvis 73%
- Katie O'Keefe 98%
- Tommy Temple 67%
- Susie Sunbeam 82%
- You 90%

Your teacher needs to know the _average_ test score.

Write a program that tells ATARI to calculate and print the average score.

**HINT:** To find the average of 5 numbers, add them together and divide by 5.

1. THINK about the problem.
2. Make your DATA TABLE here.

3. Write the ALGORITHM (steps and equations).
4. Flow Chart

5. CODE

6. DEBUG
7. REVISE
Problem 2
You are the new manager of the "Peppy Pizza" restaurant and you need the help of a computer. Write a program that will allow you to INPUT the number of small, medium, and large pizzas sold during a day.
Have ATARI print out the total number of pizzas sold and how much money you made.

PRICES: small $4.30
medium $5.50
large $7.25

OUTPUT HINT:
HOW MANY PIZZAS: (SMALL, MEDIUM, LARGE)
?  ,  ,  
THERE WERE  PIZZAS SOLD TODAY.
"PEPPY PIZZA" MADE $  .

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM
4. Flow Chart

6. DEBUG

7. REVISE
Problem 3
Write a program that will allow you to INPUT your age in years, months, and days. Example: 9 years, 3 months, 17 days.

Have ATARI calculate and print how many days, hours, and minutes old you are.

HINT: There are normally 365 days in a year and 30 days in a month. There are exactly 24 hours in a day and 60 minutes in an hour.

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM
4. Flow Chart

5. CODE

6. DEBUG
7. REVISE
Problem 4

You just got hired as a SUPER SCOOPER at the DIPPER DELIGHT Ice Cream Store. Write a program that will allow you to INPUT how many hours you worked for the week.

Have ATARI calculate and print hours worked and your salary for the week if you make $3.25 an hour.

OUTPUT HINT:
HOW MANY HOURS DID YOU WORK? _______
YOU WORKED _______ HOURS AND MADE $_____.

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM
4. Flow Chart

5. CODE

6. DEBUG
7. REVISE
Problem 5
Add to the problem you wrote for Problem 4 so that ATARI can calculate overtime pay. (Overtime is any hours worked over 40 hours a week.) You get paid $4.75 for every hour of overtime you work.

Add this to our OUTPUT:

YOU WORKED _______ OVERTIME HOURS AND
MADE $_______ IN OVERTIME.
YOUR TOTAL PAY FOR THE WEEK IS $______.
(Total pay is regular pay + overtime pay.)

HINT: You will need a decision box in your flow chart to ask:

IS H > 40?

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM
4. Flow Chart

5. CODE

6. DEBUG
7. REVISE
Problem 6

You are the famous sportscaster H.E. Nosell. You have been asked to calculate the batting averages of Big League Baseball players. Write a program that allows you to INPUT a player’s name, hits, and times at bat.

Have ATARI calculate and print the player’s name and batting average.

HINT: To calculate batting average, use this equation:

\[ \frac{1000 \times \text{hits}}{\text{times at bat}} \]

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM
Down
1. We can interact with the computer by using an ______ statement.
2. A variable that stores a number.
3. The statement used for making comparisons is ______-THEN.
4. We use ______ data to let ATARI know that we are at the end of our data list.
5. The statements that let you change your data are the ______-DATA statements.
6. A string variable is written as a letter followed by a ______ sign.
7. This sign, >, means ______ than.
8. You should document your programs by using the ______ statement.
9. A ______ table lists the variables you are using in a program.
10. The opposite of a question is called its ______.
11. An alphanumeric variable is also called a ______ variable.
11. When ATARI is alphabetizing words, it knows that "A" is the ______ letter in the alphabet.

Evaluate Yourself

1. Component 6 was ________________________________
   because ________________________________
2. The best parts of the component were ________________________________
3. The parts I liked the least were ________________________________
4. The most valuable thing I learned in this component was ________________________________
   because ________________________________
Other comments:
<table>
<thead>
<tr>
<th>Program</th>
<th>Important parts</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. 10 REM CONVERT</td>
<td>TEASPOONS TO TABLESPOONS</td>
<td></td>
</tr>
<tr>
<td>20 ? &quot;TEASPOONS&quot;, &quot;TABLESPOONS&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 FOR T = 3 TO 18</td>
<td>STEP 3</td>
<td></td>
</tr>
<tr>
<td>40 ? T, T/3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 NEXT T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60 END</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. 10 REM CONVERT</td>
<td>POUNDS TO OUNCES</td>
<td></td>
</tr>
<tr>
<td>20 ? &quot;POUNDS&quot;, &quot;OUNCES&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 FOR P = 1 TO 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 ? P, P * 16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 NEXT P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60 END</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. 10 REM CONVERT</td>
<td>YARDS TO INCHES</td>
<td></td>
</tr>
<tr>
<td>20 ? &quot;YARDS&quot;, &quot;INCHES&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 FOR Y = 1 TO 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 ? Y, Y * 36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 NEXT Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60 END</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For each conversion problem, identify the important parts by writing: HEADING, FOR–NEXT LOOP, CONVERSION EQUATION next to the lines in the program. Then write what you think ATARI would print as the output.

<table>
<thead>
<tr>
<th>Program Example:</th>
<th>Important parts</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 REM CONVERT FEET TO METERS</td>
<td>FEET</td>
<td>METERS</td>
</tr>
<tr>
<td>20 ? &quot;FEET&quot;, &quot;METERS&quot;</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>30 ?</td>
<td>2</td>
<td>0.6</td>
</tr>
<tr>
<td>40 FOR F=1 TO 10</td>
<td>FOR–NEXT LOOP</td>
<td>4</td>
</tr>
<tr>
<td>50 ? F, F*.3</td>
<td>CONVERSION</td>
<td>5</td>
</tr>
<tr>
<td>60 NEXT F</td>
<td>EQUATION</td>
<td>6</td>
</tr>
<tr>
<td>70 END</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

1. 10 REM CONVERT FEET TO YARDS
20 ? "FEET", "YARDS"
30 ?
40 FOR F=1 TO 12
   STEP 2
50 ? F, F/3
60 NEXT F
70 END
4. Convert 1–10 liters to quarts.
   CONVERSION EQUATION:
   QUARTS = L / 3.8

5. Convert 0°–100° Fahrenheit to Celsius.
   CONVERSION EQUATION:
   °C = 5*(F - 32) / 9

6. Convert 1–100 pounds to kilograms.
   CONVERSION EQUATION:
   Kilograms = P * .45
Write a conversion program for each problem. Make sure your program has a heading, FOR-NEXT loop, and conversion equation. Run your programs on ATARI to check for bugs.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Program</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Convert 1–20 inches to centimeters.</td>
<td>CONVERSION EQUATION: Centimeters = ( \text{I} \times 2.5 )</td>
</tr>
<tr>
<td>2. Convert 1–20 kilometers to miles.</td>
<td>CONVERSION EQUATION: Miles = ( \text{K} / 1.6 )</td>
</tr>
<tr>
<td>3. Convert 1–20 pounds to grams.</td>
<td>CONVERSION EQUATION: Grams = ( \text{P} \times 454 )</td>
</tr>
</tbody>
</table>
4. Flow Chart

5. CODE

6. DEBUG

7. REVISE
Use the problem-solving approach to get ATARI to solve the following conversion problems.

A. Jed needs to find out what decimal \( \frac{1}{7} \) stands for. Write a program that lists the fractions \( \frac{1}{7} \) through \( \frac{7}{7} \) and the decimals they stand for. CONVERSION EQUATION: \( \text{Decimal} = \frac{X}{7} \)

1. THINK about the problem.
2. DATA TABLE

3. ALGORITHM
4. Flow Chart

5. CODE

6. DEBUG

7. REVISE
B. Amy Astronaut is going to the moon. She learned that because the gravity on the moon is only \( \frac{1}{6} \) of the earth’s gravity, she will weigh less on the moon. Write a program that asks you to INPUT how much you weigh. Then have ATARI print how much you would weigh on the moon. CONVERSION EQUATION: moon weight = earth weight / 6

1. THINK about the problem
2. DATA TABLE

3. ALGORITHM
D. Fred's class took a test in which there were 20 questions asked. Fred's score was $16$ correct out of $20$, or $\frac{16}{20}$. Fred wants to know what percentage this would be. Write a program that lists the percentages for the test scores $\frac{1}{20}$ through $\frac{20}{20}$.

\[ \frac{X}{20} = \text{number answered correctly} \]

CONVERSION EQUATION: \[ P = \frac{X \times 100}{20} \]

1. THINK about the problem
2. DATA TABLE

3. ALGORITHM
C. Add to program #2 so that ATARI will print a conversion table after printing the output for program #2. The table should list weight on earth from 10 to 100 pounds and the equivalent moon weights.

1. Flow Chart
2. CODE
3. DEBUG
4. REVISE
CHALLENGE

E. Change program #4 so ATARI asks you to INPUT how many test questions there were (T), and how many questions you answered correctly (C). Have ATARI print your score and the percentage you got correct.

HINT: score = C out of T
      percentage = C * 100 / T

1. Flow Chart

2. CODE

3. DEBUG

4. REVISE
4. Flow Chart

5. CODE

6. DEBUG

7. REVISE
Read each RND function. Figure out what the lowest and highest random numbers will be that ATARI could print.

<table>
<thead>
<tr>
<th>Function</th>
<th>ATARI will print random numbers between and including:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>LET X = 18 * RND(1)</td>
<td>0 and 17.9999</td>
</tr>
<tr>
<td>1. LET X = 300 * RND(1)</td>
<td>___ and _____</td>
</tr>
<tr>
<td>2. LET X = RND(1)</td>
<td>___ and _____</td>
</tr>
<tr>
<td>3. LET X = 3 * RND(1)</td>
<td>___ and _____</td>
</tr>
<tr>
<td>4. LET X = 67 * RND(1)</td>
<td>___ and _____</td>
</tr>
<tr>
<td>5. LET X = 100 * RND(1) + 1</td>
<td>___ and _____</td>
</tr>
<tr>
<td>6. LET X = 25 * RND(1) + 1</td>
<td>___ and _____</td>
</tr>
<tr>
<td>7. LET X = 116 * RND(1) + 1</td>
<td>___ and _____</td>
</tr>
<tr>
<td>8. LET X = 39 * RND(1) + 1</td>
<td>___ and _____</td>
</tr>
<tr>
<td>9. LET X = 436 * RND(1)</td>
<td>___ and _____</td>
</tr>
<tr>
<td>10. LET X = 77 * RND(1) + 1</td>
<td>___ and _____</td>
</tr>
<tr>
<td>11. LET X = 43 * RND(1) + 1</td>
<td>___ and _____</td>
</tr>
<tr>
<td>12. LET X = 13 * RND(1)</td>
<td>___ and _____</td>
</tr>
<tr>
<td>13. LET X = 59 * RND(1) + 1</td>
<td>___ and _____</td>
</tr>
</tbody>
</table>
RUN the program three times on ATARI. Each time the program is run, write down the random numbers that ATARI printed. Then write the lowest and highest numbers in the list. Run the program several more times and visually note the highest and lowest numbers.

**RUN #1**

<table>
<thead>
<tr>
<th>numbers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>lowest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Program**

10 FOR L = 1 TO 5
20 LET X = 10*RND(1)
30 ? X
40 NEXT L
50 END

**RUN #2**

<table>
<thead>
<tr>
<th>numbers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>lowest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RUN #3**

<table>
<thead>
<tr>
<th>numbers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>lowest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>highest</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
We use both the INT and RND functions to tell
ATARI to print a random integer. Read each func-
tion. Then write the two numbers that ATARI must
create random integers between.

**Example:**

```
INT(40*RND(1)+26)       26   and   67

DO: 40+26+1=67
```

**ATARI will print random integers between:**

1. INT(14*RND(1)+3)   _______ and _______
2. INT(221*RND(1)+99)  _______ and _______
3. INT(3*RND(1)+2)     _______ and _______
4. INT(22*RND(1)+16)   _______ and _______
5. INT(55*RND(1)+28)   _______ and _______
6. INT(77*RND(1)+75)   _______ and _______
7. INT(94*RND(1)+33)   _______ and _______
8. INT(101*RND(1)+66)  _______ and _______
9. INT(63*RND(1)+7)    _______ and _______
10. INT(80*RND(1)+45)  _______ and _______
11. INT(46*RND(1)+23)  _______ and _______
12. INT(39*RND(1)+19)  _______ and _______
INTEGERs are whole numbers. The INT function rounds DOWN to the next whole number to make it an integer. Read each INT function. Then write what ATARI would print for the output.

<table>
<thead>
<tr>
<th>Function</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>10 LET C = 4.96</td>
<td>4</td>
</tr>
<tr>
<td>20 ? INT(C)</td>
<td></td>
</tr>
</tbody>
</table>

1. 10 LET X = 66.823  
   20 ? INT(X)  

2. ? INT(4.89)  

3. 10 LET R = 992.01  
   20 ? INT(R)  

4. ? INT(63.49321)  

5. 10 LET BD = -16.003  
   20 ? INT(BD)  

6. 10 LET P1 = 43.001  
   20 ? INT(P1)  

7. ? INT(660.666)  

8. ? INT(-33.23)  

9. 10 LET S = 4120.7  
   20 ? INT(S)  

10. ? INT(-999.999)  

150
Write an INT and RND function for each description. Remember the equation:

\[ \text{INT} \left( (B-(A+1)) \times \text{RND}(1) + A \right) \]

\(B=\text{largest number} \quad A=\text{smallest number}\)

**To print random integers between**

**Example:** 5 and 18

DO: \(\text{INT}(12 \times \text{RND}(1)+5)\)

\(\text{INT}(12 \times \text{RND}(1)+5)\)

1. 16 and 48
2. 2 and 10
3. 10 and 100
4. 1 and 50
5. 33 and 99
6. 50 and 100
7. 75 and 100
8. 27 and 41
9. 62 and 300
10. 49 and 52
Write an RND function for each description.

Create random numbers between and including:

**Example:**
0 and 9.9999

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LET $X = 10 \times \text{RND}(1)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0 and 14.9999</td>
</tr>
<tr>
<td>2.</td>
<td>0 and 92.9999</td>
</tr>
<tr>
<td>3.</td>
<td>1 and 45.9999</td>
</tr>
<tr>
<td>4.</td>
<td>0 and 70.9999</td>
</tr>
<tr>
<td>5.</td>
<td>1 and 26.9999</td>
</tr>
<tr>
<td>6.</td>
<td>0 and 106.9999</td>
</tr>
<tr>
<td>7.</td>
<td>0 and 66.9999</td>
</tr>
<tr>
<td>8.</td>
<td>1 and 211.9999</td>
</tr>
<tr>
<td>9.</td>
<td>1 and 31.9999</td>
</tr>
<tr>
<td>10.</td>
<td>1 and 441.9999</td>
</tr>
<tr>
<td>11.</td>
<td>0 and 89.9999</td>
</tr>
<tr>
<td>12.</td>
<td>1 and 53.9999</td>
</tr>
<tr>
<td>13.</td>
<td>1 and 382.9999</td>
</tr>
<tr>
<td>14.</td>
<td>0 and 554.9999</td>
</tr>
</tbody>
</table>
2. Write a CAI program that asks a student to multiply two random numbers between 1 and 10.

**Flow chart**

**Program**
Make a flow chart and write a program for each problem. Debug your programs by running them on ATARI.

1. Write a program that will print 10 random decimals between 1 and 100 and then print the integer for each.

Flow Chart

Program
1. Write your own music by adding the necessary DATA in line 60:
   10 REM WRITE A SONG
   20 READ N
   30 SOUND 1, N, 10, 8
   40 FOR T = 1 TO 150: NEXT T
   50 GOTO 20
   60 DATA

2. The above program holds the last note until you press BREAK, type END and press RETURN. Add to the above program so that it ends by itself. (HINT—use some dummy data in line 60.)
3. You can use the INPUT statement in a program with the RND and INT functions. Write a program so that ATARI will ask you to type in two integers. Then have ATARI print 10 random integers between those two numbers.

OUTPUT HINT:

TYPE IN TWO NUMBERS
AND I WILL CREATE TEN
RANDOM INTEGERS BETWEEN
THOSE TWO NUMBERS
?
10 RANDOM INTEGERS
BETWEEN AND ARE:

Flow chart  Program
1. Select a simple song from a music book (Mary Had a Little Lamb, Jingle Bells, etc.), and write a program so that the song can be played by ATARI. (HINT—Use this formula and set in the proper Notes, SOUND: 0, N, 10, 8. Later, vary the Voice, Tone, and Loudness to see what happens.)

2. Write a program so ATARI will play music that you have composed.
3. Rewrite the program so that Voice, then Tone, and then Loudness are altered. (Be careful when changing Loudness so that others around you are not disturbed.)

4. Rewrite the program so that ATARI plays Notes randomly. (HINT—drop the READ/DATA statements and use a LET statement to assign a random number to N.)
One of the most important aspects of producing good graphics, is being able to place the points and lines exactly where you want them. The key to doing so is to exactly locate a point by its column (X Coordinate) and row (Y Coordinate) position.

Use graph paper or the graphic screen illustrations that are on the next pages. Locate the following points on the Graphics 3 and Graphics 6 & 7 screens. Check your answers on ATARI. (Note — Although the two worksheet screens look similar, the numbering systems are different.)

**Graphics 3**
- 1, 10 (shown)
- 10, 1 (shown)
- 0, 0
- 5, 19
- 0, 19
- 19, 16
- 39, 0
- 39, 19
- 15, 22
- 10, 24

**Graphics 7**
- 1, 10 (shown)
- 10, 1 (shown)
- 0, 0
- 5, 19
- 0, 19
- 0, 79
- 159, 0
- 159, 79
- 60, 40
- 76, 161
- 76, 197
1. Try modifying the above program by adding COLOR statements at various lines, and by changing some of the DRAWTO and PLOT statements.

2. GET several pieces of graph paper from your teacher, or a reusable piece that has been laminated, and draw some pictures. Convert the drawings to programs, and try them on ATARI. (It's often easiest to begin with simple drawings for the Graphics 3 screen. Just be sure that you use the appropriate graph paper for the Graphics Mode you want to use.)
One of the most enjoyable aspects of graphics is the ability to draw pictures. The key to doing so is to lay out a drawing on graph paper, and then convert the graph dimensions to statements that ATARI can understand. Notice the following drawing.

Columns (X)  Graphics 3 Screen

Rows (Y)

Here's how the drawing can be programmed for ATARI to understand:

10 GRAPHICS 3
20 COLOR 3
30 PLOT 14,1
40 DRAWTO 14,10
50 DRAWTO 6,18
60 PLOT 16,1
70 DRAWTO 16,18
80 PLOT 17,1
90 DRAWTO 17,18
100 PLOT 19,1
110 DRAWTO 19,10
120 DRAWTO 27,18
130 ?“ ATARI COMPUTERS”
1. It takes a lot of practice to know all the graphic variations you can create with ATARI. Using the following program, experiment by changing the COLOR, GRAPHICS MODE, and SETCOLOR factors.

```
10 GRAPHICS 3
20 COLOR 1
30 FOR X = 0 TO 15
40 SETCOLOR 0, X, 2
50 PLOT 5, 5
60 DRAWTO 25, 5
70 FOR T = 1 TO 600: NEXT T
80 ?? ?? ?? X
90 NEXT X
100 END
```

2. Take some graphic programs you have already written, or make some new ones, and improve them by using the SETCOLOR statement.
One enjoyable aspect of graphics is animation—causing the graphics to move. Following is a program for some simple animation.

```
10 GRAPHICS 3
20 COLOR 1
30 FOR X=0 TO 39 STEP 3
35 ? #6,"" ESC SHIFT CLEAR ""
40 PLOT X,7
50 DRAWTO X,10
60 DRAWTO X+3,10
70 DRAWTO X+3,7
80 DRAWTO X,7
85 FOR T=1 TO 100: NEXT T
87 IF X>=36 THEN GOTO 10
90 NEXT X
```

Here's what the program does. Lines 40 through 80 make a simple square graphic. The X Coordinate is not specified in these lines, but rather it is set as a variable X. Lines 30 and 90 make the X Coordinate as every third number between 0 and 39. These lines, along with 10 and 20, which determine the graphics mode and color, are the main part of this program. However, notice how the program was improved by adding some more lines after the program was first written. Line 85 is a "timer" so that the graphic remains momentarily on the screen. (Try changing this line for different effects.) Line 87 causes the program to repeat once the graphic has moved completely across the screen. Line 35 causes the screen to be cleared as the graphic starts over. (Remember, #6 must be used with a PRINT statement for the graphics screen!)

1. Run this program, and then modify some line statements to see how you can change the graphics and animation.
2. Use all the graphic techniques that you have learned to this point to make your own animated graphics.
Use what you know about a good game program to write the game programs described below.

1. It is more meaningful to the user when the computer calls him or her by name—it makes the interaction more personal.
   Write a GUESS A NUMBER game program that asks the user's name and calls the user by name throughout the program.

1. THINK about the program
2. DATA TABLE

3. ALGORITHM
Using all the techniques you have learned for graphics, sound, and regular programming, create a fantastic light and sound show!
2. Revise the game program in #1 so a player must answer "YES" or "NO" in line 150.

If anything else is typed for INPUT, make ATARI print the question in LINE 140 again. This helps make the program GOOF PROOF.
4. Flow Chart

5. CODE

6. DEBUG

7. REVISE
4. Add something to the game program in #3 so ATARI asks the user the top number in the range they wish to guess. (For example, 1 to ____)?

After the user types in the top number, use it in the RND function to create a random integer between 1 and the top number.

Hint: IF N = the top number

THEN you would use this RND function:

LET X = INT(N*RND(1)+1)
3. Add something to the game program in #2 so ATARI tells the user how many tries it took before they guessed the correct number.

HINT: Use a COUNTER, C.

Set C at 0 before the first guess. After the first guess add 1 to the counter: \( C = C + 1 \).

Then after each of the next guesses, make sure one more is added to the counter. When the correct number is guessed, make ATARI print IT TOOK YOU ______ GUESSES.
4. Flow Chart

5. CODE

6. DEBUG
7. REVISE
5. Make up a computer game that uses a die. Write a program using all of the good style techniques you’ve learned.

The RND function for the throw of your die must choose a random integer between 1 and 6.

**Example**

1. THINK about the program.
2. DATA TABLE

3. ALGORITHM
4. Flow Chart

5. CODE

6. DEBUG

7. REVISE
6. Create a computer program for any game you like. Include the five things every good game program should have. Try using animation. Be creative!

1. THINK about the program
2. DATA TABLE

3. ALGORITHM
Down
1. X and Y positions are called _______.
3. Write your program to be user-______.
5. A word meaning "having no pattern or specific purpose:" _______.
7. C. stands for the ______ statement.
9. "Convert" means to _______.
11. The function which creates whole numbers in a program: _______.
13. This stands for a random function: _______.

Across
2. The statement which tells ATARI to place a point on the screen at a certain location: _______.
4. GR. stands for the ___________ statement.
6. The text _______ makes up the lower four lines of the screen in Graphics Mode.
8. We use this command when we want to hear ATARI: _______.
10. The statement which tells ATARI to connect two points is _______.
12. If we want to change the color of our graphics or screen we must use the _______ statement.
14. Another name for a whole number is _______.

Evaluate Yourself
1. Component 7 was ____________________________________ because ____________________________________

2. The best parts of the component were ____________________________________

3. The parts I liked the least were ____________________________________

4. The most valuable thing I learned in this component was ____________________________________ because ____________________________________

Other comments:
COMPONENT 7 FUN PAGE

Word Bank

CHANGE  INTEGER
COLOR  PLOT
COORDINATES  RANDOM
DRAWTO  RND
FRIENDLY  SETCOLOR
GRAPHICS  SOUND
INT  WINDOW

178
Instant (Freeze-Dried Computer Programming in) BASIC—2nd Astounding! Edition
Jerald R. Brown

Here is an active, easy, painless way to learn BASIC. This primer and workbook gives you a fast, working familiarity with the real basics of BASIC. It is one of the smoothest and best-tested instructional sequences going!

ISBN 0-918398-57-6 200 pages $12.95

Are You Computer Literate?
Karen Billings and David Moursund

This is a fun introduction to computers for the real novice. Written by educators, this is a book that teaches the capabilities, limitations, applications and implications of computers.

ISBN 0-918398-29-0 160 pages $9.95

An Apple® For Kids / A PET® For Kids
Sharon Boren

A fresh, instructive, fun approach to teaching kids programming and computer operation. Written by a teacher, these books teach good programming skills with a focus on problem solving, improved thinking skills and creativity.

Classroom Activity Workbook and Teachers Manual also available.

Computers, Teaching and Learning
Jerry Willis, D. Lamont Johnson and Paul Dixon

For the educator interested in using computers or the computer-user interested in educational applications, this book covers most of the important topics of concern to the educational environment.

ISBN 0-88056-065-7 200 pages $9.95

BRAINFOOD — Our catalog listing of over 130 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.
dilithium Press books are available at your local bookstore or computer store. If there is not a bookstore or computer store in your area, charge your order on VISA or MC by calling our toll-free number, (800) 547-1842.

Send to: dilithium Press, P.O. Box E, Beaverton, OR 97075
Please send me the book(s) I have checked. I understand that if I’m not fully satisfied I can return the book(s) within 10 days for full and prompt refund.

☐ Instant (Freeze-Dried Computer Programming in) BASIC—2nd Astounding! Edition
☐ Are You Computer Literate?

☐ □ Check enclosed $_________
   Payable to dilithium Press

☐ □ Please charge my
   VISA □ MASTERCHARGE □
   # __________________ Exp. Date __________

Name ________________________________
Address ________________________________
City, State, Zip ____________________
Signature ________________________________

☐ Send me your catalog, Brainfood.
<table>
<thead>
<tr>
<th></th>
<th>Yes, send me your free catalog, BRAIN FOOD, which lists over 150 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NAME ____________________________________________________________</td>
</tr>
<tr>
<td></td>
<td>ADDRESS _________________________________________________________</td>
</tr>
<tr>
<td></td>
<td>CITY, STATE, ZIP _______________________________________________</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Yes, send me your free catalog, BRAIN FOOD, which lists over 150 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NAME ____________________________________________________________</td>
</tr>
<tr>
<td></td>
<td>ADDRESS _________________________________________________________</td>
</tr>
<tr>
<td></td>
<td>CITY, STATE, ZIP _______________________________________________</td>
</tr>
</tbody>
</table>

Mail To:
dilithium Press
P.O. Box E
Beaverton, OR 97075

Or Call:
800-547-1842
This workbook is an excellent tool for teaching children how to program a microcomputer in BASIC. It's an individualized, self-paced approach that accompanies the book *AN ATARI FOR KIDS*.

Most activities in this workbook can be done without a computer, so can be used as seatwork (solving the problem of 1 computer and 25 students).

Written approximately at the 4th grade reading level, this workbook:

- Includes worksheets to go along with each chapter of *AN ATARI FOR KIDS*
- Approaches programming in a fresh, fun way
- Is easy for kids to understand
- Contains illustrations interesting to kids
- Provides practice and reinforcement for programming skills your students learn
- Has activities that focus on problem solving, improved thinking skills and creativity
- Concludes each unit with fun pages that test your students' learning progress

*AN ATARI FOR KIDS* and

*AN ATARI IN THE CLASSROOM:*
Teacher's Guide also available.