

CHAPTER 17

EXTRA RESOURCES

Additional Resources

1. “Array Basics” (<http://tiny.cc/arraybasics/>): Head to this site for more examples of what you can do with arrays.
2. Array Review and Examples (<http://tiny.cc/arrayreview/>): Review arrays and dig into more examples of arrays.
3. “Sprite Arrays” (<http://tiny.cc/spritearrays/>): Find out how to store sprites in arrays and when you’d want to use sprite arrays.

Review Questions

1. How many subscripts do you need to access an element in a two-dimensional array?
2. How many subscripts do you need to access an element in a three-dimensional array?
3. How many rows and columns are there in a 4×5 matrix?

4. Which element of the score matrix would each of the following expressions return (from which row and column)?
 - a. `score[3][4]`
 - b. `score[5][1]`
5. True or False? To zero out all elements of the 3×3 `mat` matrix, you write `mat = 0`.
6. What is the syntax for accessing the element in the fifth row and second column of the `mat` matrix?
7. Give the dimensions of the following matrix:

1	2	0
6	4	3

Practice Exercises

1. What does the `mat` matrix contain after running the following code?

```

For I = 1 To 4
  For J = 1 To 4
    mat[I][J] = I * J
  EndFor
EndFor

```

2. What does the `mat` matrix contain after running the following code?

```

For I = 1 To 4
  For J = 1 To 4
    If (I > J) Then
      mat[I][J] = 0
    Else
      mat[I][J] = 1
    EndIf
  EndFor
EndFor

```

3. Modify the previous program to generate the following matrix.

0	1	1	1	1
-1	0	1	1	1
-1	-1	0	1	1
-1	-1	-1	0	1
-1	-1	-1	-1	0

4. Find the error in the following program segment:

```
' Find the sum of each row of the matrix
sum = 0
For I = 1 To 4
  For J = 1 To 4
    sum = sum + mat[I][J]
  EndFor
  TextWindow.WriteLine("Sum of row " + I + " = " + sum)
EndFor
```

5. DreamPants uses Small Basic to keep track of its pants inventory. Its program contains a matrix named `stock` that stores its stock of pants with waist sizes of 28 to 40 inches and lengths from 28 to 40 inches. Write a subroutine that fills the `stock` matrix with random numbers from 0 to 10 to represent the number of pants DreamPants has in stock for each category (waist size versus length).
6. Write a program that fills up a 4×4 matrix named `score` with numbers entered by the user.
- Write a subroutine that finds the lowest number in `score`.
 - Write a subroutine that finds the average of each row in `score`.
 - Write a subroutine that finds the average of each column in `score`.
 - Write a subroutine that finds the products of the entries in the upper-left to the lower-right diagonal of `score`.
 - Write a subroutine that computes the average of all the elements of `score`.
 - Write a subroutine that finds the largest element on each diagonal.
7. Write a program that simulates throwing two dice. The program should do the following:
- Zero out all the elements of the 6×6 `roll` matrix.
 - Start a loop that runs 36,000 times. In each iteration of the loop, get two random numbers from 1 to 6 (one for the first die and the other for the second die). Then increment the element in `roll` that corresponds to the pair of numbers you obtained. For example, if the first die was a 2 and the second die was a 4, increment `roll[2][4]` by 1.
 - When the loop terminates, print out the `roll` matrix.
 - Describe the result of your simulation and how it works.

Here's a sample output:

	1	2	3	4	5	6
1	1003	991	950	1001	984	1019
2	982	1025	942	969	1048	1072
3	1084	1013	986	1025	1020	996
4	962	1013	1025	993	990	960
5	982	1021	1015	1032	959	968
6	1042	945	1020	1006	1000	957

8. Look at the following table of distances between cities (in miles):

	Atlanta	Boston	Chicago	Dallas
Atlanta	0	1095	715	805
Boston	1095	0	983	1815
Chicago	715	983	0	931
Dallas	805	1815	931	0

Write a program that asks the user to input two cities from this table and then displays the distance between them. Here is a sample output of this program:

```
Enter first city: Dallas
Enter second city: Chicago
The distance between Dallas and Chicago is 931
```

```
Enter first city: Detroit
Enter second city: Boston
Sorry! I don't know this.
```

9. The following table shows part of a teacher's attendance sheet. The numbers represent the days each student was absent in each of the four school terms.

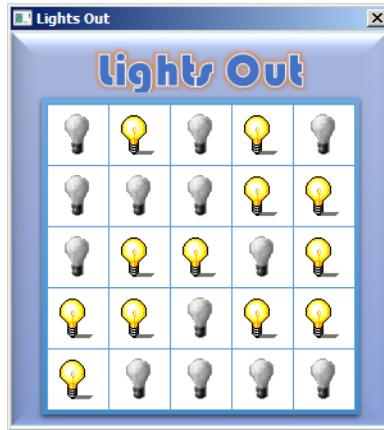
	Term1	Term2	Term3	Term4
Jenny	0	1	1	2
Jake	2	3	0	2
Karen	3	0	1	1

Write a program that prompts the user to enter a student's name and the term number (from 1 to 4) and then displays the number of days the student was absent in that term.

10. Write a program that creates and prints a matrix in the following form, where the numbers represented by a , b , c , and d in the first row are input by the user.

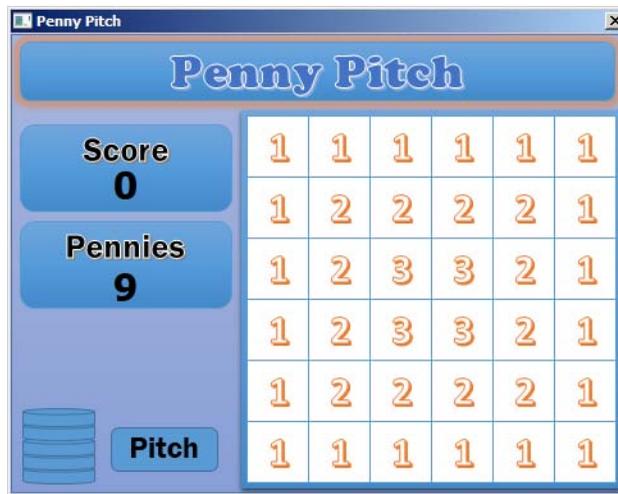
$$\begin{array}{cccc}
 a & b & c & d \\
 a+1 & b-1 & c \times 2 & d / 2 \\
 a+2 & b-2 & c \times 4 & d / 4 \\
 a+3 & b-3 & c \times 8 & d / 8
 \end{array}$$

11. Open the file *LightsOut_Incomplete.sb* in this chapter's folder. The game contains a 5×5 grid of lights that can be either on or off (see the following figure). Clicking on a light toggles that light and the four surrounding nondiagonal lights (that is, up, down, left, and right). The goal of the game is to turn all the lights off.



Your task is to complete the `ProcessClick()` subroutine, which is called when the player clicks any lightbulb in the grid.

- Open the file `PennyPitch_Incomplete.sb` in this chapter's folder. The user interface for this game is shown in the following figure.



The user starts with nine pennies. When the player clicks the **Pitch** button, one penny is tossed onto a random square on the game board. The player's score goes up based on the number that the penny lands on. But there's a catch! Five squares on the board have hidden bombs on them. If the tossed penny lands on any of these squares, the player loses the game.

Complete the `NewGame()` subroutine and fill the 6×6 board array at the start of each game. You also need to write the code that resets all the elements of this array to zero and then sets five random elements to 1 (to represent the bomb locations).

