

## CHAPTER 15

# EXTRA RESOURCES

### Additional Resources

1. “Array Basics” (<http://tiny.cc/arraybasics/>): Head to this article for more examples of what you can do with arrays.
2. “Indexing Operator” (<http://tiny.cc/indexingoperator/>): Learn more about the indexing operator in Small Basic arrays.
3. Array Review and Examples (<http://tiny.cc/arrayreview/>): Review arrays and dig into more examples of arrays.
4. “Unconventional Indexing of Arrays” (<http://tiny.cc/unconventional/>): See the different options for indexing your array elements.
5. “Empty Elements” (<http://tiny.cc/emptyelements/>): Read about how to remove an element from an array.
6. “Dice Simulation” (<http://tiny.cc/dicesim/>): Follow along with this robust tutorial and dig deeper into this chapter’s dice simulation Programming Challenge.
7. “Prime Sieve” (<http://tiny.cc/prim sieve/>): Follow this tutorial through a program that finds prime numbers.

8. “Pinball Simulation” (<http://tiny.cc/pinball/>): Get more information about this chapter’s pinball simulation Programming Challenge.
9. “Flower Anatomy Quiz” (<http://tiny.cc/flowerquiz/>): Expand on this chapter’s flower quiz Programming Challenge using these details and tutorials.

## Review Questions

1. What is an array?
2. When should you use arrays in your programs?
3. What is an element?
4. What is the difference between indexed arrays and associative arrays?
5. What is a string initializer?
6. Create an array named `score` that contains these three numbers: 10, 15, and 20. Indexing starts at one.
7. What notation do you use to refer to the elements of an array?

Create an array called `arr` that contains `[2 -2 8 -10 3 6]` with indexing beginning at one. Use this array to answer questions 8–14 about the `arr` array, which is a pirate’s favorite array name!

8. What is the value of each of the following?
  - a. `arr[2]`
  - b. `arr[5 - 2]`
  - c. `2 * arr[3]`
  - d. `arr[4] * arr[4]`
  - e. `arr[arr[5]]`
  - f. `3 + arr[6]`
  - g. `arr[4] + arr[5]`
  - h. `arr[arr[3] / arr[1]]`
  - i. `Math.Max(arr[2], arr[4])`
  - j. `Math.Min(arr[1], arr[3])`
9. Let `x = 2` and `y = 3`. Find the value of each of the following.
  - a. `arr[x] * arr[y]`
  - b. `arr[x * y]`
  - c. `arr[x + y]`
  - d. `arr[2 * y / x]`
  - e. `arr[arr[y - x]]`
  - f. `arr[y + x]`

10. Write a For loop that assigns the number 5 to all elements of arr.
11. Write a For loop that displays all the values of arr on a single line with commas between the numbers.
12. Write a For loop that displays every other element of arr (the second, fourth, sixth, and so on).
13. Write a For loop that displays the elements of arr in reverse order.
14. Write a For loop that copies the positive elements of arr into a new array, posArr.
15. Create an array num that contains these 10 elements: 6 7 3 4 5 4 3 2 1 0. Begin indexing at one. Write a Small Basic statement or a For loop to do each of the following:
  - a. Assign the value 8 to the third element.
  - b. Display the value of a randomly selected element.
  - c. Display the sum of the last two elements.
  - d. Display the index of the maximum element.
  - e. Display the sum of all the elements.
16. What's the output of this program?

---

```

tree[1] = "Palm"
tree[2] = "Maple"
tree[3] = "Oak"
tree[4] = "Chestnut"
M = 2
N = 1
TextWindow.WriteLine(tree[M])
TextWindow.WriteLine(tree[2 * M - N])
TextWindow.WriteLine(tree[N] + " Tree")

```

---

17. What are the contents of the arr2 array after executing the following program?

---

```

arr1 = "1=10;2=15;3=20;4=35;5=62"
For N = 1 To 5
    arr2[N] = arr1[6 - N]
    TextWindow.WriteLine(arr2[N])
EndFor

```

---

18. Find the error in the following program and fix it.

---

```

For N = 1 To 10
    score[J] = Math.GetRandomNumber(100)
EndFor

```

---

- What's wrong with the following program? Fix the error, and then add a For loop to display the five names input by the user in reverse order.

---

```

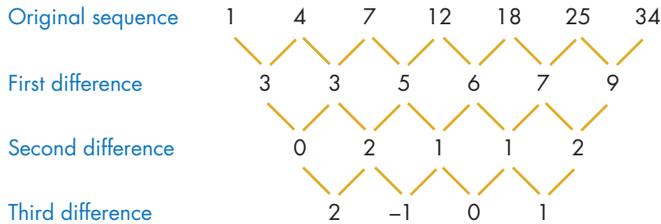
For N = 1 To 5
    TextWindow.Write("Enter name # " + N + ": ")
    name(N) = TextWindow.Read()
EndFor

```

---

## Practice Exercises

- In Olympic high diving, each of the seven judges records a score from one to ten. Write a program that reads the scores of the seven judges and stores them in an array named score. Then have your program discard the lowest and highest scores, and display the average of the remaining five scores.
- Write a program that reads seven numbers from the user. Calculate and display the differences between the consecutive terms. Then calculate the differences between those numbers. Perform this calculation and display the results one more time so that you have calculated three sets of numbers, as shown in the following figure:



- The game Reverse starts by displaying nine digits (1 to 9) in random order. The game then asks the player how many digits (counting from the left) to reverse. For example, if the sequence is 1 3 2 8 5 6 9 7 4 and the player enters 4 (to reverse the first four digits), the game displays 8 2 3 1 5 6 9 7 4. The goal of the game is to arrange the list in ascending order from left to right (1 2 3 4 5 6 7 8 9). Write a program to implement this game.
- Write a program that generates ten unique random integers between 1 and 100, and stores them in an array named rand.
- Write a program that reads ten numbers from a user (allow the user to enter a number more than once). Display the list with the duplicate numbers removed. For example, if the input list was 1 2 1 2 3 4 5 4 3 6, your program should display 1 2 3 4 5 6.
- Write a program that simulates a lucky number lottery. The lottery tickets have numbers from 1 to 1000. Create an array that holds the 30 winning numbers. Elements 1 to 10 hold numbers that win \$100. Elements 11 to 20 hold numbers that win \$50, and elements 21 to 30

hold numbers that win \$25. Then draw a lottery ticket at random and decide whether that number is a winner. If the drawn ticket is a winner, display the amount won.

7. Write a program that evaluates a polynomial (such as  $3 + 2x - 5x^2 + 3x^3$ ) for different values of  $x$ . Store the coefficients of the polynomial in an array (for example,  $[3 \ 2 \ -5 \ 3]$ ). When a user enters a value for  $x$ , create a powers array like this:  $[1 \ x \ x^2 \ x^3]$ . Then use a For loop to multiply the coefficients array with the powers array, and keep track of the cumulative sum.
8. Write a program that stores ten positive or negative numbers (entered by the user) in an array. Then write a For loop that calculates the absolute values of the elements of the array and displays the largest number.
9. Write a program that reads five numbers from the user and stores them in an array (named `arr`) as follows. Assign the first value to `arr[1]` and `arr[10]`. Assign the second value to `arr[2]` and `arr[9]`, and so on.
10. The elements of an array (`arr`) are related to each other as follows:

---

$$\text{arr}[N] = \text{arr}[N-1] + \text{arr}[N-2]$$

---

Use this formula in your program so that it reads `arr[1]` and `arr[2]` from the user and then creates array elements 3 to 20.

11. Five candidates are nominated to the positions of president and vice president in your class. Write a program that reads the names of the five candidates in an array (`name`). Then display all the possible pairs (president, vice president).
12. Write a program that has two arrays that save employees names and their pay rate per hour. Then have your program display a table like this:

---

Employee#	Name	Pay rate/hour
1	Jack	\$9.5
2	Helda	\$12.0
...		

---

Prompt the user to enter an employee number and how many hours they worked in a week. Have your program compute and display the employee's total pay.

13. A class has 15 students. Their grade letters (A, B, C, D) are saved in an array named `grade`. Write a program that tallies the grades and shows the total number of students who got an A, B, C, or D.
14. A school has 25 lockers numbered 1 to 25. A student opens every locker. A second student closes every other locker, beginning with locker 2. A third student changes every third locker, beginning with locker 3 (they open the closed lockers and close the open lockers). A fourth student changes every fourth locker, beginning with locker 4, and so on. Which lockers are open after the 25th student completes their pass? Write a

program to find the answer. Here's a sample output of the program that shows the status of the lockers after the fourth student completes their pass (a 1 means the locker is open and a 0 means it's closed).

---

```

                1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
-----
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1
1 0 0 1 1 1 1 1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 1 0 1

```

---

15. You are trapped in a cave that has ten rooms. Each room has four doors: three closed doors and one open door. Run this program to find out how long it takes you to escape the cave. Explain how the program works.

---

```

For N = 1 To 10 ' 10 rooms
    door[N] = Math.GetRandomNumber(4) ' The number of the open door
EndFor

room = 1
tries = 0
While (room <= 10)
    selDoor = 0 ' The door selected by player
    While (selDoor <> door[room])
        tries = tries + 1
        TextWindow.WriteLine("You're now in room " + room + ".")
        TextWindow.Write("Which door do you want to try (1-4)? ")
        selDoor = TextWindow.ReadNumber()
        If (selDoor = door[room]) Then
            TextWindow.WriteLine("This door is open.")
        Else
            TextWindow.WriteLine("This door is closed.")
        EndIf
        TextWindow.WriteLine("")
    EndWhile
    room = room + 1 ' Goes to the next room
EndWhile

TextWindow.WriteLine("CONGRATULATIONS!")
TextWindow.WriteLine("You escaped the cave in " + tries + " tries!")

```

---