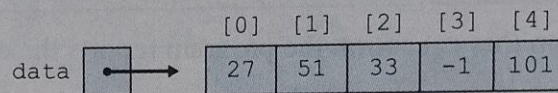


Self-Check Problems**Section 7.1: Array Basics**

- Which of the following is the correct syntax to declare an array of ten integers?
 - `int a[10] = new int[10];`
 - `int[10] a = new int[10];`
 - `[]int a = [10]int;`
 - `int a[10];`
 - `int[] a = new int[10];`
- What expression should be used to access the first element of an array of integers called `numbers`? What expression should be used to access the last element of `numbers`, assuming it contains 10 elements? What expression can be used to access its last element, regardless of its length?
- Write code that creates an array of integers named `data` of size 5 with the following contents:



- Write code that stores all odd numbers between -6 and 38 into an array using a loop. Make the array's size exactly large enough to store the numbers.

Then, try generalizing your code so that it will work for any minimum and maximum values, not just -6 and 38.

- What elements does the array `numbers` contain after the following code is executed?

```
int[] numbers = new int[8];
numbers[1] = 4;
numbers[4] = 99;
numbers[7] = 2;

int x = numbers[1];
numbers[x] = 44;
numbers[numbers[7]] = 11; // uses numbers[7] as index
```

- What elements does the array `data` contain after the following code is executed?

```
int[] data = new int[8];
data[0] = 3;
data[7] = -18;
data[4] = 5;
data[1] = data[0];

int x = data[4];
data[4] = 6;
data[x] = data[0] * data[1];
```


7. What is wrong with the following code?

```
int[] first = new int[2];
first[0] = 3;
first[1] = 7;
int[] second = new int[2];
second[0] = 3;
second[1] = 7;

// print the array elements
System.out.println(first);
System.out.println(second);

// see if the elements are the same
if (first == second) {
    System.out.println("They contain the same elements.");
} else {
    System.out.println("The elements are different.");
}
```

8. Which of the following is the correct syntax to declare an array of the given six integer values?

- a. `int[] a = {17, -3, 42, 5, 9, 28};`
- b. `int a {17, -3, 42, 5, 9, 28};`
- c. `int[] a = new int[6] {17, -3, 42, 5, 9, 28};`
- d. `int[6] a = {17, -3, 42, 5, 9, 28};`
- e. `int[] a = int [17, -3, 42, 5, 9, 28] {6};`

9. Write a piece of code that declares an array called `data` with the elements 7, -1, 13, 24, and 6. Use only one statement to initialize the array.
10. Write a piece of code that examines an array of integers and reports the maximum value in the array. Consider putting your code into a method called `max` that accepts the array as a parameter and returns the maximum value. Assume that the array contains at least one element.
11. Write a method called `average` that computes the average (arithmetic mean) of all elements in an array of integers and returns the answer as a `double`. For example, if the array passed contains the values [1, -2, 4, -4, 9, -6, 16, -8, 25, -10], the calculated average should be 2.5. Your method accepts an array of integers as its parameter and returns the average.

Section 7.2: Array-Traversal Algorithms

12. What is an array traversal? Give an example of a problem that can be solved by traversing an array.
13. Write code that uses a `for` loop to print each element of an array named `data` that contains five integers:

```
element [0] is 14
element [1] is 5
element [2] is 27
element [3] is -3
element [4] is 2598
```

Consider generalizing your code so that it will work on an array of any size.

14. What elements does the array `list` contain after the following code is executed?

```
int[] list = {2, 18, 6, -4, 5, 1};
for (int i = 0; i < list.length; i++) {
    list[i] = list[i] + (list[i] / list[0]);
}
```

15. Write a piece of code that prints an array of integers in reverse order, in the same format as the `print` method from Section 7.2. Consider putting your code into a method called `printBackwards` that accepts the array as a parameter.
16. Describe the modifications that would be necessary to change the `count` and `equals` methods developed in Section 7.2 to process arrays of `Strings` instead of arrays of integers.
17. Write a method called `allLess` that accepts two arrays of integers and returns `true` if each element in the first array is less than the element at the same index in the second array. Your method should return `false` if the arrays are not the same length.

Section 7.3: Reference Semantics

18. Why does a method to swap two array elements work correctly when a method to swap two integer values does not?
19. What is the output of the following program?

```
public class ReferenceMystery1 {
    public static void main(String[] args) {
        int x = 0;
        int[] a = new int[4];
        x = x + 1;
        mystery(x, a);
        System.out.println(x + " " + Arrays.toString(a));
        x = x + 1;
        mystery(x, a);
        System.out.println(x + " " + Arrays.toString(a));
    }
    public static void mystery(int x, int[] a) {
        x = x + 1;
        a[x] = a[x] + 1;
        System.out.println(x + " " + Arrays.toString(a));
    }
}
```

20. What is the output of the following program?

```
public class ReferenceMystery2 {
    public static void main(String[] args) {
        int x = 1;
        int[] a = new int[2];
        mystery(x, a);
        System.out.println(x + " " + Arrays.toString(a));
        x--;
        a[1] = a.length;
        mystery(x, a);
    }
}
```



```

        system.out.println(x + " " + Arrays.toString(a));
    }

    public static void mystery(int x, int[] list) {
        list[x]++;
        x++;
        system.out.println(x + " " + Arrays.toString(list));
    }
}

```

21. Write a method called `swapPairs` that accepts an array of integers and swaps the elements at adjacent indexes. That is, elements 0 and 1 are swapped, elements 2 and 3 are swapped, and so on. If the array has an odd length, the final element should be left unmodified. For example, the list [10, 20, 30, 40, 50] should become [20, 10, 40, 30, 50] after a call to your method.

Section 7.4: Advanced Array Techniques

22. What are the values of the elements in the array `numbers` after the following code is executed?

```

int[] numbers = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
for (int i = 0; i < 9; i++) {
    numbers[i] = numbers[i + 1];
}

```

23. What are the values of the elements in the array `numbers` after the following code is executed?

```

int[] numbers = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
for (int i = 1; i < 10; i++) {
    numbers[i] = numbers[i - 1];
}

```

24. Consider the following method, `mystery`:

```

public static void mystery(int[] a, int[] b) {
    for (int i = 0; i < a.length; i++) {
        a[i] += b[b.length - 1 - i];
    }
}

```

What are the values of the elements in array `a1` after the following code executes?

```

int[] a1 = {1, 3, 5, 7, 9};
int[] a2 = {1, 4, 9, 16, 25};
mystery(a1, a2);

```

25. Consider the following method, `mystery2`:

```

public static void mystery2(int[] a, int[] b) {
    for (int i = 0; i < a.length; i++) {
        a[i] = a[2 * i % a.length] - b[3 * i % b.length];
    }
}

```