Self-Check Problems

Section 7.1: Array Basics

1. Which of the following is the correct syntax to declare an array of ten integers?

   a. `int a[10] = new int[10];`

   b. `int[10] a = new int[10];`

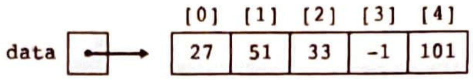
   c. `[]int a = [10]int;`

   d. `int a[10];`

   [✓ e.] `int[] a = new int[10];`

2. 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? numbers[0] numbers[9] numbers[numbers.length - 1]

3. Write code that creates an array of integers named `data` of size 5 with the following contents:



int[] data = {27, 51, 33, -1, 101};

4. 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. It didn't say inclusive or exclusive, so I did what was easiest, inclusive min & exclusive max

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

```
public static int[] range (int min, int max) {
   int[] r = new int[max - min];
   for(int i = 0; i < r.length; i++) r[i] = i + min;
   return r;
}
```

5. 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
```

[0, 4, 11, 0, 44, 0, 0, 2];

6. 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];
```

[3, 3, 0, 0, 6, 9, 0, -18]

```
pubic static double avg (int[] ints) {
    int sum =0;
    for (int i : ints) sum += i;
    return sum / ints.length;
}
```

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) { // The arrays contain the same values, but are not the same. To compare
    System.out.println("They contain the same elements.");// values you must use. Arrays.equals()
} 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};`

```
public static int max (int[] arr) {
    int[] arrCopy = Arrays.copyOf(arr, arr.length);
    Arrays.sort(arrCopy);
    return arrCopy[arrCopy.length -1];
}
```

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. `int[] data = {7, -1, 13, 24, 6};`

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.
    Accessing every element of the array sequentially. It can ~~solve~~ be used to find the sum of elements

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
```

```
for (int i = 0; i < arr.length; i++) {
    SOPf("element [%d] is %d %n", i, arr[i]);
}
```

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?

[3, 24, 8, -5, 6, 1]

```
int[] list = {2, 18, 6, -4, 5, 1};
for (int i = 0; i < list.length; i++) {
    list[i] = list[i] + (list[i] / list[0]); // list[i] *= ~~3/2.0~~; 4.0/3
}
```

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. replace arr.length with str.length(), replace arr[i] with str.charAt(i);

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?

```
public static void printBackwards (int[] arr) {
    SOP("[");
    if (arr.length > 0) SOP(arr[0]);
    for (int i = 1; i < arr.length; i++) SOP(", " + arr[0]);
}
```

```
public static void printBackwards (int[] arr1, int[] arr2) {
    if (arr1.length != arr2.length) return false;
    for (int i = 0; i < arr1.length; i++) {
        if (arr1[i] >= arr2[i]) return false;
    }
    return true)
}
```

19. What is the output of the following program?

```
public class ReferenceMystery1 {
    public static void main(String[] args) {
        int x = 0;   // x = 0
        int[] a = new int[4]; // a = [0, 0, 0, 0]
        x = x + 1; // x = 1
        mystery(x, a); // x = 1, a = [0, 0, 1, 0]
        System.out.println(x + " " + Arrays.toString(a));
        x = x + 1; // x = 2
        mystery(x, a);
        System.out.println(x + " " + Arrays.toString(a));
    }
    public static void mystery(int x, int[] a) {
        x = x + 1; // x = ~~2~~ 3
        a[x] = a[x] + 1; // a = [0, 0, 1, 1]
        System.out.println(x + " " + Arrays.toString(a));
    }
}
```

```
2 [0, 0, 1, 0]
1 [0, 0, 1, 0]
3 [0, 0, 1, 1]
2 [0, 0, 1, 1]
```

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]; // [0, 0]
        mystery(x, a); // x = 1, a = [0, 1]
        System.out.println(x + " " + Arrays.toString(a));
        x--; // 0
        a[1] = a.length; // [0, 4]
        mystery(x, a);
```

```
2 [0, 1]
1 [0, 1]
1 [1, 4]
0 [1, 4]
```

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

    public static void mystery(int x, int[] list) {
        list[x]++; // [0, 1]  ]
        x++; // 2
        System.out.println(x + " " + Arrays.toString(list));
    }
}
```

```
public static void swapPairs (int[] a) {
    for (int i = 0; i < a.length / 2; i++) {
        final int temp = a [2 * i];
        a[2 * i] = a[2 * i + 1];
        a[2 * i + 1] = temp;
    }
}
```

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];
}
```

[20, 30, 40, 50, 60, 70, 80, 90, 100, 100]

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];
}
```

[10, 10, 20, 30, 40, 50, 60, 70, 80, 90]

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];
    }
}
```

[26, 19, 14, 11, 10]

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];
    }
}
```

[1, 3, -3, 13, -4, -24, -8, -14]

0 1 2 3 4 5 6 7

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

```
                0   1  2  3  4   5   6   7
int[] a1 = {2, 4, 6, 8, 10, 12, 14, 16};
int[] a2 = {1, 1, 2, 3, 5, 8, 13, 21};
             0  1  2  3  4  5   6   7
mystery2(a1, a2);
```

26. Consider the following method, `mystery3`:

```
public static void mystery3(int[] data, int x, int y) {
    data[data[x]] = data[y];
    data[y] = x;
}
```

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

```
int[] numbers = {3, 7, 1, 0, 25, 4, 18, –1, 5};
mystery3(numbers, 3, 1);// [7, 3, 1, 0, 25, 4, 18, -1, 5]
mystery3(numbers, 5, 6);// [7, 3, 1, 0, 4, 5, 18, -1, 5]
mystery3(numbers, 8, 4);// [7, 3, 1, 0, 8, 4, 18, -1, 5]
```

27. Consider the following method:

```
public static int mystery4(int[] list) {
    int x = 0;
    for (int i = 1; i < list.length; i++) {
        int y = list[i] - list[0];
        if (y > x) {
            x = y;
        }
    }
    return x;
}
```

What value does the method return when passed each of the following arrays?

a. `{5}`// 0

b. `{3, 12}`// 9

c. `{4, 2, 10, 8}` // 6

d. `{1, 9, 3, 5, 7}` // 8

e. `{8, 2, 10, 4, 10, 9}`// 2

28. Consider the following method:

```
public static void mystery5(int[] nums) {
    for (int i = 0; i < nums.length - 1; i++) {
        if (nums[i] > nums[i + 1]) {
            nums[i + 1]++;
        }
    }
}
```

What are the final contents of each of the following arrays if each is passed to the above method?

```
public static double avglength (String[] a) {
  int l = 0;
  for (String str: a) l += str.length;
  return (double) l / a.length;
}
```

```
public static boolean sc30 (String[] arr) {
  String[] reverse = new String[arr.length];
  for(int i = 0; i < arr.length; i++)
    reverse[reverse.length-1-i] = arr[i];
  return Arrays.equals(arr, reverse);
}
```

a. {8} // [8]

b. {14, 7} // [14, 8]

c. {7, 1, 3, 2, 0, 4} // [7, 2, 3, 3, 1, 4]

d. {10, 8, 9, 5, 5} // [10, 9, 9, 6 6]

e. {12, 11, 10, 10, 8, 7} // [12, 12, 11, 11, 9, 8]

29. Write a piece of code that computes the average `String` length of the elements of an array of `Strings`. For example, if the array contains `{"belt", "hat", "jelly", "bubble gum"}`, the average length is `5.5`.

30. Write code that accepts an array of `Strings` as its parameter and indicates whether that array is a palindrome—that is, whether it reads the same forward as backward. For example, the array `{"alpha", "beta", "gamma", "delta", "gamma", "beta", "alpha"}` is a palindrome.

Section 7.5: Multidimensional Arrays

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

```
int[][] numbers = new int[3][4];
for (int r = 0; r < numbers.length; r++) {
    for (int c = 0; c < numbers[0].length; c++) {
        numbers[r][c] = r + c;
    }
}
```

[
[0, 1, 2, 3]
[1, 2, 3, 4]
[2, 3, 4, 5]
]

32. Assume that a two-dimensional rectangular array of integers called `data` has been declared with four rows and seven columns. Write a loop to initialize the third row of data to store the numbers 1 through 7.

```
for (int i = 0; i < 7; i++) data[2][i] = i+1;
```

33. Write a piece of code that constructs a two-dimensional array of integers with 5 rows and 10 columns. Fill the array with a multiplication table, so that array element `[i][j]` contains the value `i * j`. Use nested `for` loops to build the array.

```
int[][] table = new int[5][10];
for (int r=0; r < table.length; r++) {
  for (int c=0; c < table[r].length; c++) {
    table[r][c] = r*c;
  }
}
```

34. Assume that a two-dimensional rectangular array of integers called `matrix` has been declared with six rows and eight columns. Write a loop to copy the contents of the second column into the fifth column.

```
matrix[5] = matrix[2];
/*If matrix is int[][], then there is no reason to
  use a loop to copy element by element, as primitive
  are not references, so this works fine */
```

35. Consider the following method:

```
public static void mystery2d(int[][] a) {
    for (int r = 0; r < a.length; r++) {
        for (int c = 0; c < a[0].length - 1; c++) {
            if (a[r][c + 1] > a[r][c]) {
                a[r][c] = a[r][c + 1];
            }
        }
    }
}
```

If a two-dimensional array `numbers` is initialized to store the following integers, what are its contents after the call shown?

```
int[][] numbers = {{3, 4, 5, 6},
                   {4, 5, 6, 7},
                   {5, 6, 7, 8}};
mystery2d(numbers);
```

[[4, 5, 6, 6]
[5, 6, 7, 7]
[6, 7, 8, 8]]

36. Write a piece of code that constructs a jagged two-dimensional array of integers with five rows and an increasing number of columns in each row, such that the first row has one column, the second row has two, the third has three, and so on. The array elements should have increasing values in top-to-bottom, left-to-right order (also called row-major order). In other words, the array's contents should be the following:

```
1
2, 3
4, 5, 6
7, 8, 9, 10
11, 12, 13, 14, 15
```

Use nested `for` loops to build the array.

```
int [][] m = new int[5][]; int i = 1;
for (int r = 0; r < m.length; r++) {
   m[r] = rew int[r + 1];
   for (int c = 0; c < m[r].length; c++) m[r][c] = i++;
}
```

Exercises

1. Write a method called `lastIndexOf` that accepts an array of integers and an integer value as its parameters and returns the last index at which the value occurs in the array. The method should return –1 if the value is not found. For example, in the array `[74, 85, 102, 99, 101, 85, 56]`, the last index of the value `85` is 5.

2. Write a method called `range` that returns the range of values in an array of integers. The range is defined as 1 more than the difference between the maximum and minimum values in the array. For example, if an array called `list` contains the values `[36, 12, 25, 19, 46, 31, 22]`, the call of `range(list)` should return `35` (46 − 12 + 1). You may assume that the array has at least one element.

3. Write a method called `countInRange` that accepts an array of integers, a minimum value, and a maximum value as parameters and returns the count of how many elements from the array fall between the minimum and maximum (inclusive). For example, in the array `[14, 1, 22, 17, 36, 7, –43, 5]`, for minimum value `4` and maximum value `17`, there are four elements whose values fall between `4` and `17`.

4. Write a method called `isSorted` that accepts an array of real numbers as a parameter and returns `true` if the list is in sorted (nondecreasing) order and `false` otherwise. For example, if arrays named `list1` and `list2` store `[16.1, 12.3, 22.2, 14.4]` and `[1.5, 4.3, 7.0, 19.5, 25.1, 46.2]` respectively, the calls `isSorted(list1)` and `isSorted(list2)` should return `false` and `true` respectively. Assume the array has at least one element. A one-element array is considered to be sorted.

5. Write a method called `mode` that returns the most frequently occurring element of an array of integers. Assume that the array has at least one element and that every element in the array has a value between `0` and `100` inclusive. Break ties by choosing the lower value. For example, if the array passed contains the values `[27, 15, 15, 11, 27]`, your method should return `15`. (*Hint*: You may wish to look at the `Tally` program from this chapter to get an idea how to solve this problem.) Can you write a version of this method that does not rely on the values being between `0` and `100`?

6. Write a method called `stdev` that returns the standard deviation of an array of integers. Standard deviation is computed by taking the square root of the sum of the squares of the differences between each element and the mean, divided by one less than the number of elements. (It's just that simple!) More concisely and mathematically, the standard deviation of an array *a* is written as follows:

$$stdev(a) = \sqrt{\frac{\sum_{i=0}^{a.length-1} (a[i] - average(a)^2)}{a.length - 1}}$$