**Unit 8:** Arrays

Answer the questions and complete the programs in preparation for the end of Unit exam

**Review Questions:**

1. Consider the following array:  
   int[] a = { 1, 2, 3, 4, 5, 4, 3, 2, 1, 0 };

What are the contents of the array a after the following loops complete?

1. for(int i =1; i < 10; i++){ a[i] = a[i – 1];}

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

1. for(int i =9; i > 0; i--){ a[i] = a[i – 1];}
2. for(int i =0; i < 9; i++){ a[i] = a[i + 1];}
3. for(int i =8; i >= 0; i--){ a[i] = a[i + 1];}
4. for(int i =1; i < 10; i++){ a[i] = a[i] + a[i - 1];}
5. for(int i =1; i < 10; i = i + 2){ a[i] = 0;}
6. for(int i =0; i < 5; i++){ a[i + 5] = a[i];}
7. for(int i =1; i < 5; i++){ a[i] = a[9 - i];}
8. Write the Java code for a loop that simultaneously computes both maximum and minimum values in an array.
9. A run is a sequence of adjacent repeated values. Give pseudocode for computing the length of the longest run in a array. For example, the longest run in the array with elements  
    1 2 5 5 3 1 2 4 3 2 2 2 2 3 6 5 5 6 3 1  
   has a length of 4.
10. What is wrong with the following method that aims to fill an array with random numbers?

public void makeCombination(int[] values, int n)

{

Random generator = new Random();

int[] numbers = new int[values.length];  
 for( int i = 0; i < numbers.length; i++)

{

numbers[i] = generator.nextInt(n);

}

values = numbers;

}

**Exercises:**

1. Write array methods that carry out the following tasks for an array of integers by completing the ArrayMethods class below. For each method, provide a test program.

public class ArrayMethods{

private int[] values;  
 public ArrayMethods(int[] initialValues) { values = initialValues; }  
 public void swapFirstAndLast(){ …. }  
 public void shiftRight() { … }  
 …..  
}

* 1. Swap the first and last element in the array.
  2. Shift all elements by one to the right and move the last element into the first.
  3. Replace all even elements with 0.
  4. Replace each element except the first and last by the larger of its two neighbors.
  5. Remove the middle elements if the array length is odd, or the middle two elements if the length is even.
  6. Move all even elements to the front, otherwise preserve the order of the elements.
  7. Remove the second-largest element in the array.
  8. Return true if the array is currently sorted in increasing order.
  9. Return true if the array contains two adjacent duplicate elements.
  10. Return true if the array contains duplicate elements (which need not be adjacent)

1. *Magic squares*. An *n x n* matrix that is filled with numbers 1,2,3,…,n^2 is a magic square if the sum of the elements in each row, in each column, and in the two diagonals is the same value.  
   Write a program that reads in 16 values from the keyboard and tests whether they form a magic square when put into a 4 x 4 array.

|  |  |  |  |
| --- | --- | --- | --- |
| 16 | 3 | 2 | 13 |
| 5 | 10 | 11 | 8 |
| 9 | 6 | 7 | 12 |
| 4 | 15 | 14 | 1 |