WestVirginiaUniversity
INSTITUTE OF TECHNOLOGY

Computer Science 121 – Fall 2019

Project 5: One-dimensional Array Operations

Deadline: Monday, December 9, (midnight) in e-campus.

- **Objectives:** This assignment will test your understanding of one-dimensional arrays using method and various basic java syntax.
- **Class name:** Your file needs to be named as **YourLastNameProject5.java**
- **Description:** In this project, you will be working with arrays. Follow the instructions carefully to get exact output as shown in the samples. You program must also work correctly for various test cases used for grading.
  You are advised to read the question completely first. Instead of writing all the methods at once and testing, build it one by one and test. You will need to partially write code for the main method at step 7 first. After that you can use method in step 1 and 2 to test your code. Then keep adding other methods and testing your output for accuracy.

1. Name of method: **readNumbers**

   Formal input parameter: int[] arr

   Return type: void

The method will read integers from the user through console using Scanner and assign it to the array. Input will be provided on a single line with each number separated by a space. In this method, there is only one input parameter which is the array.

2. Here you will create two printNumbers methods using method overloading
   - Name of method: **printNumbers**

     Formal input parameter: int[] arr

     Return type: void

There is only one input parameter which is the array. The array elements will be printed inside curly braces separated by commas. For example, if you have integers 1, 2, 3, 4 in the array, you should print it as {1, 2, 3, 4}. Observe how there is no comma at the end.

   - Name of method: **printNumbers**

     Formal input parameter: boolean[] arr

     Return type: void

There is only one input parameter which is the array. The array elements will be printed inside curly braces separated by commas. For example, if you have boolean values true, false, true, false in the array, you should print it as {true, false, true, false}. Observe how there is no comma at the end.

Both methods in Q2 must have exactly the same name printNumbers and must not return anything.

3. Name of method: **analyzeNumbers**
   Formal input parameter: int[] arr
   Return type: int

The method will first find the average of the numbers in the array, then count how many numbers are above the average and return it. There is only one input parameter which is the array, and it returns an integer value.

4. Name of method: **compareMax**
   Formal input parameter: int[] arr
   Return type: boolean[]retArr

This method will first find the maximum element in the array. You need to create a new temporary array here with the same size as arr. For each element in the input array, arr, if the element is greater than the half of the maximum element (the integer value, no need to cast to double), you need to set the Boolean value corresponding to that index in the temporary array as true, otherwise it will be set as false.

For example, you have an array arr containing elements {1 ,2, 3, 4, 5}, then the max is 5 and integer half of max is 2

Then you need to return array retArr containing elements {false, false, true, true, true}

5. Name of method: **moveValuesRight**
   Formal input parameter: int[] arr
   Return type: void

This method will move the elements one position to the right in the array itself, and the right most element is moved to the front. For example, for the array {1, 2, 3, 4} the method moves the elements as {4, 1, 2, 3}. The size of the array must not change, and the method should overwrite the original array and not return anything.

The method will have only one input parameter which is the array to be modified.

6. Name of method: **selectDistinct**
   Formal input parameter: int[] arr
   Return type: int[]

This method will return an array consisting of only the distinct elements from the original array. You will need to preserve only the first occurrence of each element and maintain the order. For example, for the array {1,2,3,2,1,6,3,4,5} the method will need to return {1,2,3,6,4,5}. (This example of array is only for representation, you basically need to return an array containing those elements)

There is only one input parameter which is the array and the return type is also an array.

7. In the main method,
   - You will first ask the user how many elements the user wants to enter. If the user enters a value less than or equal to 0, then you need to ask the question again.

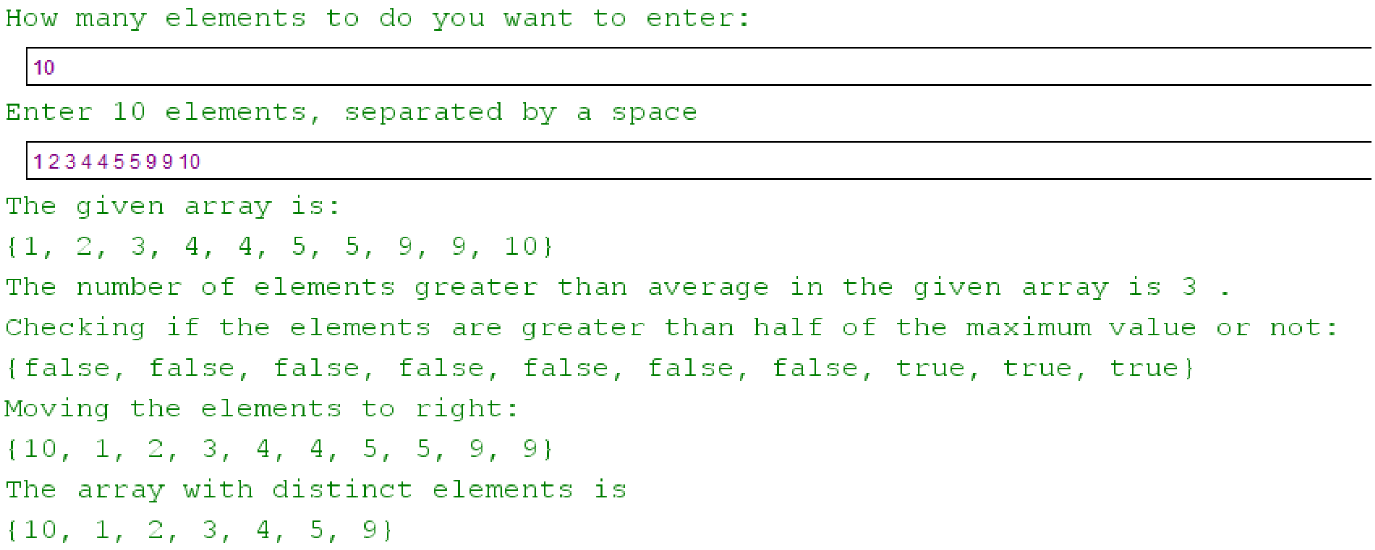
- Based on that you will create the array in the main method. (The main array must be created in main method and no other methods.)
- After that, use the readNumbers method to get numbers from the user.
- Then print the array elements using printNumbers method. Use the prompt “The given array is:” before printing the numbers.
- Then find the number of elements which are greater than the average in the array using analyzeNumbers method. Print the count using the prompt “The number of elements greater than average in the given array is”
- Then use the method compareMax to get the array which shows if each element is greater than half of the maximum value in the array.
- Then print the array elements using printNumbers method. Use the prompt “Checking if the elements are greater than half of the maximum value or not:” before printing the numbers.
- Then use method moveValuesRight to move the elements one right (last element at the first).
- Then print the resulting array (original array) using prompt” Moving the elements to right:”
- Then use selectDistinct method to find the array containing only the distinct elements form the original array.
- Print the array containing distinct elements using printNumbers method. Use the prompt “The array with distinct elements is” before displaying.

These instructions must be done sequentially in order. A program displaying output of sequence will be penalized (even if each output is correct).

➢ **Submission:** Upload .java file through your campus account (ecampus.wvu.edu).

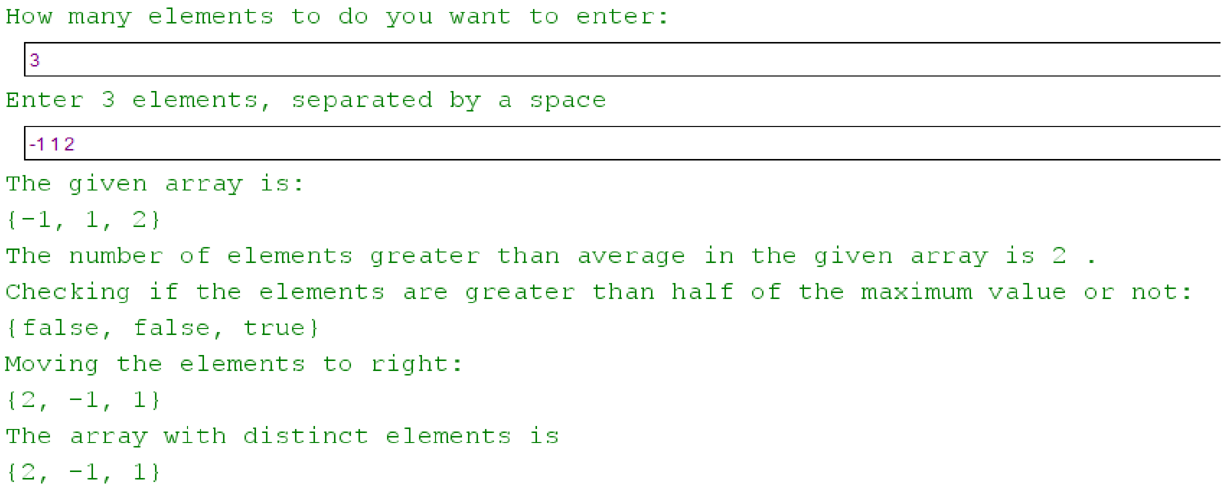
- **Screenshot:** Here are some example screenshots of the output:

Example 1



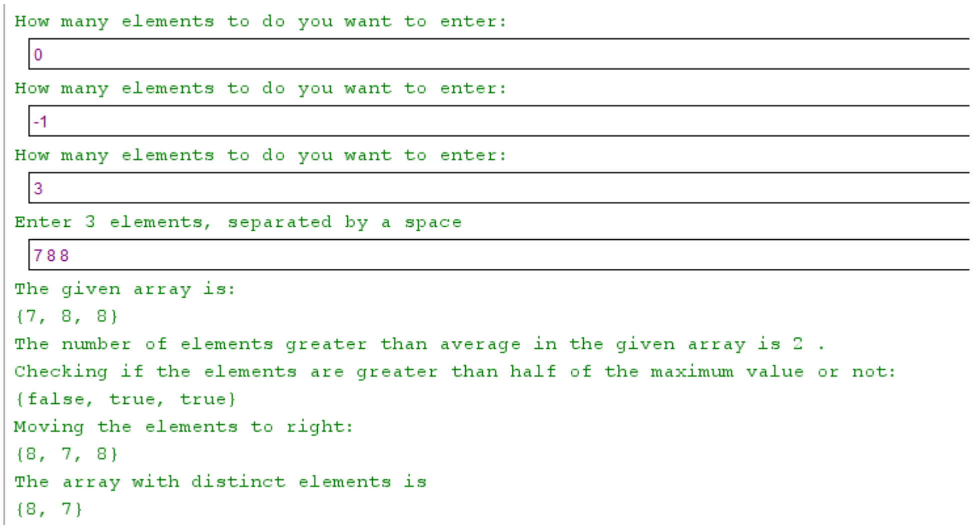
```
How many elements to do you want to enter:
 10
Enter 10 elements, separated by a space
 1 2 3 4 4 5 5 9 9 10
The given array is:
{1, 2, 3, 4, 4, 5, 5, 9, 9, 10}
The number of elements greater than average in the given array is 3 .
Checking if the elements are greater than half of the maximum value or not:
{false, false, false, false, false, false, false, true, true, true}
Moving the elements to right:
{10, 1, 2, 3, 4, 4, 5, 5, 9, 9}
The array with distinct elements is
{10, 1, 2, 3, 4, 5, 9}
```

Example 2



```
How many elements to do you want to enter:
 3
Enter 3 elements, separated by a space
 -1 1 2
The given array is:
{-1, 1, 2}
The number of elements greater than average in the given array is 2 .
Checking if the elements are greater than half of the maximum value or not:
{false, false, true}
Moving the elements to right:
{2, -1, 1}
The array with distinct elements is
{2, -1, 1}
```

Example 3



```
How many elements to do you want to enter:
0
How many elements to do you want to enter:
-1
How many elements to do you want to enter:
3
Enter 3 elements, separated by a space
788
The given array is:
{7, 8, 8}
The number of elements greater than average in the given array is 2 .
Checking if the elements are greater than half of the maximum value or not:
{false, true, true}
Moving the elements to right:
{8, 7, 8}
The array with distinct elements is
{8, 7}
```

Grading Criteria:

The grading criteria is tentative, and instructors might choose to change it if needed.

Tentative Grading criteria	Max
Comments at the beginning and throughout the program	5
Java syntax is correct (class names starting with capital letter, variable names starting with small letter, methods starting with small letter etc)	5
Readability (indentations, meaningful variable naming)	5
Efficiency (There is no redundant code, loops used properly, etc)	5
readNumbers performs as expected	10
Overloaded methods, printNumbers method performs as expected	15
analyzeNumbers method performs as expected	10
compareMax method performs as expected	10
moveValuesRight method performs as expected	10
main method performs as expected	10
Program execution as expected	10
Output format is as expected	5
Total	100
Comments	

Note: If your program doesn't compile, the program will not be graded and get 0.