**Lab 22 – ArrayAnalyzer**

Open BlueJ, and create a new BlueJ project titled **Lab22-ArrayAnalyzer** in your CS\LABS folder.

Create a new class and **type** in our code skeleton:

//Name:

import java.util.\*;

public class PracticeProblems

{

public static void main(String[] args)

{

Scanner console = new Scanner(System.in);

**int[] nums = {2, 4, 0, 8, 6, 2, 3, 5, 9, 0, 9, 6};**

}

}

An array is a single variable (of particular type) that stores a **list** of values. The way we access / change values is by referring to the **index** (or position) of the value in the array. We use ‘square bracket notation’ to refer to elements in an array. For an array called *nums*, the value of the element in the first position would be *nums[0]* (Java, like most languages, starts index numbers at 0). Arrays can be confusing, but they are too useful not to use!

As always, more information is available in our notes.

**Before each problem, insert a COMMENT with the problem number. Note that your code should work no matter what values *nums* has. No hard-coding! I will change your array b+++efore grading!**

1. Note the bolded addition to our skeleton code above. Add this variable declaration to your program – the variable nums will be used to test the code you write in the following problems. Print the value of the last element in numsto the console. **(Remember that I might test your code with a different array that might have a different length.)**
2. Using the lengthproperty of an array, print the number of elements in nums to the console.
3. Write a for loop that will *iterate* through all elements in nums and print each value, separated by a space. (Do not use Arrays.toString().)
4. Print the sum of all elements in nums to the console.
5. Print “Contains an odd num” to the console (only once) if an odd number is found in nums*.*  If no numbers are odd, print “Odd-free!”.

//a boolean variable (used as a flag) will help for these type of problems

1. Print “No zeros” to the console if nums DOES NOT contain a 0. Print “Found a zero” (once!) if nums contains a 0.
2. (Riddle) There are 3 ants, sitting on the three corners of a triangle. All ants randomly pick a direction and start moving along edge of the triangle. What is the probability that any two ants collide?
3. Print “Plenty of 6s” if numscontains at least two sixes. Print “Less than two 6s” if not.
4. Print “Found 2-3 combo” if, in nums, there is a 2 **immediately** followed by a 3. Print “No 2-3 combo” otherwise.
5. Print “Three in a row” if three consecutive elements in numshave the same value. Print “No three in a row” if not.
6. Complete the “Worksheet - Arrays” Excel file. Save when finished.

**Array Analyzer app**

Create a new class and **type** in our code skeleton:

//Name:

import java.util.\*;

public class ArrayAnalyzer

{

public static void main(String[] args)

{

Scanner console = new Scanner(System.in);

}

}

Your program should get an array of doubles from the user, and then give some statistics about that array.

First, you will need to ask the user how many numbers they plan to enter. Use that number to size your array. Then use a loop to gather the values and fill the array.

Once all numbers have been put into the array, print out these stats about the array:

1. What is the largest number?
2. What is the smallest number?
3. What is the average of all the numbers?
4. How many of the numbers are even when looking only at the whole number portion?
5. What is the longest segment of repeating numbers?
6. What is the longest segment of increasing numbers?
7. What is the longest segment of decreasing numbers?

Sample input/output. (Input is in **RED**)

How many numbers will you enter? >>> **13**

Type in numbers with spaces in between >>> **4.5 9.2 3 5 6 7 4.3 -2 8 8 8 3 3**

Largest number: 9.2

Smallest number: -2.0

Average (rounded to tenth): 5.2

Number of Evens: 7

Longest repeat: 8.0 is repeated 3 times

Longest increasing run: 3.0 5.0 6.0 7.0

Longest decreasing run: 7.0 4.3 -2.0