CSC 152 Lab 13 – Arrays in-class exercises (REVIEW)

** You are to implement the CSC 152 in-class exercise from Tuesday 4/9. You can use previous labs and/or notes**

Program Name: **LastnameFirstinitial_Lab13.java**
Comment

Name: YOURNAME
Section: Section [S1/S2] / Today date
"I, YOURNAME, pledge to follow the Honor Code in taking my Lab 13."

Screenshot filename: LastnameFirstinitial_Lab13Output.docx

Your program MUST compile without any errors. **Any compile error will be -30 points.**

**** Make sure to create structure and have everything works in this program ****

Everything will be in main method. NO ADDITIONAL METHODs added. So, when you do the search, you do not have a method to use, so you MUST write the code to search each element in the array as we discussed

Part A: Question 1 from In-Class

1. Partially Filled Integer Array – **Code Segment** (50 points)
   a. Create an integer array size 30, called **scores**
   b. You will ask the user to enter an integer and store in **scores** as many as they want. They can stop when they enter -1, or when they fill up the array.
      BUT nothing should be negative number.
   c. You will print the numbers only the user entered, 10 numbers per line, each number separated by a tab.
   d. At the end, you will print how many score entered, the total scores entered, highest scores entered, and the lowest score entered.

Part B: Question 2 from In-Class

2. Write a **code segment** for **a sequential/linear search** on array **scores** (previous questions; note that it is partially filled, so you need to make sure to get the correct upper boundary check)
(50 points)
   a. Ask the user for a **target** number to search.
   b. Write **a sequential/linear search on the array and once you finish your search, you will print if the target number is found or not in the array**. NO METHOD
   c. Create a new Array called **sortedScores** with the size of the same number that the user entered in partially filled array scores (question 1)
   d. Copy the content from scores to **sortedScores**
   e. Called **Arrays.sort(sortedScores)** // this will sort the number

f. Write another part for a binary search NO METHOD (using **sortedScore** array – assume already sorted data entered from above). NO METHOD

** Run the program and make sure that you have output screenshots for each section **

Once complete, submit your ***.java and screenshot .docx** files on Blackboard
You must submit what you have completed during the lab hours.