Mr. Bellavia

EXERCISE 02 – 1D ARRAY METHODS

IMPORTANT: Before submission, make a copy of your **'Program.cs'** file for each question and then rename each file to the following:

File Names:

- *last_name_first name*_U4_E02_1.cs
- *last_name_first name*_U4_E02_2.cs
- *last_name_first name*_U4_E02_3.cs
- *last_name_first name*_U4_E02_4.cs
- *last_name_first name*_U4_E02_5.cs
- *last_name_first name*_U4_E02_6.txt

Note: Along with last name and first name, make sure the end of the filename (i.e., before the **.cs**) has the **unit number**, **exercise number**, and **question number.** For example:

smith_*john*_***U1_E03_2***.cs

1.

a. Write a program that asks the user for 5 integers, then output those integers in ascending order.

b. Output the same integers in descending order. **Hint:** Look up the Array.Sort() method for C# arrays to find out how to sort in descending order.

2. Create a string array of colours (i.e., blue, green, red, etc.). Ask the user to input a colour, then check and tell the user if that colour exists in the array. **Hint:** Make use of the Array.IndexOf() method (look up examples on the internet!).

3. Create the following .csv file:

input.csv:

```
5
Doe,Jane
88,89,91,95
Smith,John
45,76,56,34
Sacramanto,Bob
75,81,83,68
Matthews,Auston
91,85,82,93
Smith,Jane
34,51,62,70
```

- The first line is an integer N representing the number of records in the file.
- The following N pairs of lines are records where the first line of a pair consists of the last name and first name of a student, and the second line of a pair consists of 4 marks for that student.

Write a program that reads in the above input file. As each record is read in (i.e., using a for-loop), use the .Split() method to store the last name and first name in a string array, and the marks in a double array. Also, as you are reading each record, output the min, max, and average of each record. Finally, output the name and average of the student with the highest average (you can assume that no average is ever the same for this question). **Note:** You will need to use .Split() for the marks into a string array first, then create a separate double array and use a for-loop to copy and convert the marks from the string array into the double array.

4. **[THINK]** Create and **populate** an array of 5 doubles. Ask the user to input an index 'i'. Your program should delete the value at index 'i' in the array by shifting all elements to the left and making the last element equal to '0'. Lastly, output a sorted version of this new array. For example:

   Array of 5 doubles:

3.11	8.54	6	**3.2**	**12.98**
0	1	2	3	4

   User inputs 2 for 'i' which results in (before sorting):

3.11	8.54	**3.2**	**12.98**	**0**
0	1	2	3	4

   Hint: Inside your for-loop you can access the next element by adding 1 to the index, for example: myArray**[x + 1]**

5. **[OPTIONAL] [THINK]** Consider the following array of size 20 that contain integers between 1 and 10:

```
int[] myInts = { 3, 5, 6, 9, 10, 10, 3, 7, 5, 4, 2, 1, 2, 4, 3, 7, 8, 5, 2, 6 };
```

   Write a program that outputs all the duplicates that exist in the array. For example:

   Sample **Output:**

```
2
3
4
5
6
7
10
```

   Hint: You should first sort your array. Also, you will probably want to make use of a nested for-loop.
   Note: This question might drive you a little crazy 😊

6. **[OPTIONAL]** Research the difference between Array.BinarySearch() and Array.IndexOf(). Give a scenario of when you should use each (you can save your answer in a **.txt** file).