1) Introduction to Arrays (with Debugging)

In the first part of this lab, we would like to introduce you to arrays, parameter passing, and the more uses of the Debugger.

b) Run the program using input 4, -7, 29, 8, 34

<mark>What is the syntax error and how did you fix it?</mark>

```
// Display the array contents
Dump();
```

When we are calling the method Dump() , there are no default values and so we need to pass the values in the method

Correction:

```
Dump(dataArray,n);
```

<mark>What is the output?</mark>

Enter the number of elements (between 0 and 10) => 4
Enter the Element 0 => -7
Enter the Element 1 => 29
Enter the Element 2 => 8
Enter the Element 3 => 34
The elements in the array are:
 -7
 29
 8
 34

c) Run the program using input -2, 15, 4, -7, 29, 8, 34

<mark>What is the output?</mark>

Enter the number of elements (between 0 and 10) => -2
Enter the number of elements (between 0 and 10) => 15
Enter the number of elements (between 0 and 10) => 4
Enter the Element 0 => -7
Enter the Element 1 => 29

Enter the Element 2 => 8
Enter the Element 3 => 34
The elements in the array are:
 -7
29
 8
34

 We have a do-while loop which says to continue the loop till the number of elements in the array entered by the user should be in between 0 ad 10. So, when we entered -2, the compiler didn't exit the loop and we had o enter the number of elements again. Then we entered 15 which was again greater than 10, so not acceptable and we had to enter again.
Next we entered 4, which is correct (in between 0 and 10), so we entered the same elements as we entered in part(b), hence, output is same.

e) Now run the program (F5) using the input of 4, -7, 29, 8, 34.

What are the values in `arrValues` and `num` in *Fill*?

Name	Value	Type
▲ arrValues	{int[10]}	int[]
[0]	9	int
[1]	8	int
[2]	7	int
[3]	6	int
[4]	5	int
[5]	4	int
[6]	3	int
[7]	2	int
[8]	1	int
[9]	0	int
num	0	int

2

f) Press (F5) to resume your program from the current instruction and notice that it provides you the opportunity now to input the data.

Name	Value	Type
◢ dataArray	{int[10]}	int[]
[0]	-7	int
[1]	29	int
[2]	8	int
[3]	34	int
[4]	5	int
[5]	4	int
[6]	3	int
[7]	2	int
[8]	1	int
[9]	0	int
n	4	int

Why have they changed?

We passed 'n' as reference variable, and not passed by value, thus the variable changes its value because the value entered in the method was stored in the memory location, so the value changes in the Main() method as well.

The array was passed as an entire array reference, and not passed by value, so the values entered by user entered in the memory and thus the values changes in the Main() method as well.

g) Press (F11) once again and notice the program has jumped into the method `Dump`.

What are the values in `arrVals` and `num` in *Dump?*

Name	Value	Type
◢ arrVals	{int[10]}	int[]
[0]	-7	int
[1]	29	int
[2]	8	int
[3]	34	int

[4]	5	int
[5]	4	int
[6]	3	int
[7]	2	int
[8]	1	int
[9]	0	int
num	4	int

h) Single step (F11) your way through *Dump* until at least 2 values from the array are printed on the output screen and then press (F5) to move to the end of the program.

<mark>What is the output?</mark>

Enter the number of elements (between 0 and 10) => 4
Enter the Element 0 => -7
Enter the Element 1 => 29
Enter the Element 2 => 8
Enter the Element 3 => 34
The elements in the array are:
 -7
29
8
34

<mark>Why are only 4 values from the array being output given that you know from Part (g) that there were 10 values in the array that was to be printed?</mark>

Only 4 values from the array are printed, the values entered by the user and not all 10 values in array because, we asked the user in Fill() Method to enter a number between 0 and 10, the number of values he wishes to enter in array, and user enters 4. Since the value was passed by reference, the value of num (num=4) is passed back to n, so n=4.
While calling Dump() Method, we pass n by value, and n=4. The loop used to print the values of array has end point 'n' which is equal to 4 (n<4) and not 10 and so the number of values of array printed are 4 and not 10.

i) Add *ReverseDump*

<mark>Give the details of the *ReverseDump* method.</mark>

```
// Method:        ReverseDump
```

```
// Description:  Display the contents of an array in reverse order.
// Parameters:   arrItems: array to be displayed in reverse order.
//               num: number of elements in the array.
// Returns:      void
public static void ReverseDump(int[] arrItems, int num)
{
    //output values from array in reverse order
    Console.WriteLine("The elements in array in reverse order are:");
    for (int i = num - 1; i >= 0; --i)
        Console.WriteLine(" {0} ", arrItems[i]);
}
```

For loop, we start with i=num-1 because that is the index of the array to be printed first(the last position) and it goes on till i=0.

Give the details of the *ReverseDump* call (in *Main*).

```
//Display array in reverse order
 ReverseDump(dataArray,n);
```

We call ReverseDump() function and pass the array in which user entered the values and the number of elements entered by the user.

What is the output?

Enter the number of elements (between 0 and 10) => 4
Enter the Element 0 => -7
Enter the Element 1 => 28
Enter the Element 2 => 8
Enter the Element 3 => 34
The elements in the array are:
 -7
 29
 8
 34
The elements in array in reverse order are:
 34
 8
 29
 -7

Answer all the highlighted questions in a file and then submit a PDF of this file (called it Lab4_1.pdf) to the Lab 4 dropbox. When asked "What is the output", simply type in what is seen in the output window.

2) Putting it All Together

5

Demonstrate to the Lab personnel and Submit the .cs file online