Lottery Program

This document will support the portfolio assessment by providing helpful hints and tips to complete the tasks. The program brief is an application that runs a lottery program. Users can choose from a series of numbers between a particular range. The program then creates its own version of numbers and compares those user values against the program's random numbers. Any matches that the program then finds are then displayed to the user.

Task 1 – Taking user input and storing values

- User input can be gathered by using:

```
Console.ReadLine()
```

- This method returns a string value so store it as a variable like this:

```
string userInput = Console.ReadLine();
```

- Convert a string into another type by calling the types “parse” or “tryparse” methods. Check the signature of these methods to better understand the inputs and outputs.
- Create an array using full syntax since we don’t have the values yet. *(Note: size can be dynamic by using a variable instead making it easier to change later).*
- Loops can be used to encapsulate this code so we can enter multiple values into the array easily. Remember a for loops syntax:

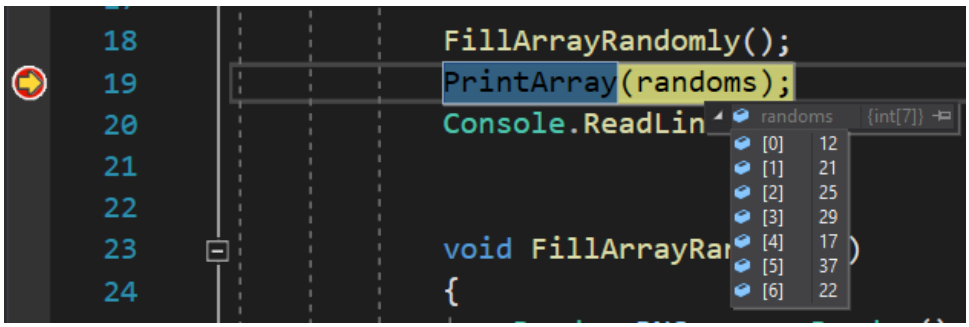
```
For(int I = 0; I < myarray.Length; I++)
{


}
```

Task 2 – Debugging

Create a screenshot of your application in debugging/breakpoint mode and show the values for the array that the user entered using the above task.

- Enter break mode by using the bar on the left of visual studio.



- Screen capture the arrays data by hovering over the variable in the IDE.

Task 3 – Random Number Generator

- Create a random number generator object and number using this code:

```
Random rnd = new Random();

Int randomNumber = rnd.next(lowestValue, highestValue);
```

- Create a loop to enter values into an array randomly.

Task 4 – Demonstrate two methods of searching an array

For this task you need to create two search methods of finding a value in an array. Make sure you create this code as a function that is reusable.

- Remember function declarations include return type, name and any parameters in brackets:

```
Void print(string theMessage)

{



}
```

- Use the supporting power point on sorting and searching for help on how to perform binary searches.
- Remember to return a result if the value is not found!

Task 5 – Complete the program

Tie the following four tasks together and complete the program by using the input, random generation and searching to finalise the lottery application. Tidy up user outputs, validate user inputs and check the additional tasks if you want an extra challenge.

Task 6 – Test and Debug your application

Using debugging tools and running your application in debug mode, validate that the application runs as expected and does what the initial brief asked.