

SOFTWARE DEVELOPMENT

Computing



Reka Ragupathy
RAG16831638

Table of Contents

1. Exercise 1
 - a. Coding
 - b. List of problems
 - c. Program screenshots
 - d. Description of tests
2. Exercise 2
 - a. Coding
 - b. List of problems
 - c. Program screenshots
 - d. Description of tests

No design documentation
Prior to coding.
See short paragraph at end
& code

1. Exercise 1

a) Coding

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
//REKA RAGUPATHY
```

```
namespace TwoPlayerGame
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            int[] playerOneEntries = new int[7];
```

```
            int[] playerTwoEntries = new int[7];
```

```
            string result = "";
```

```
            bool playerTwoFailsOnce = false;
```

```
            Console.WriteLine("Player one please enter 7 values: ");
```

```
            for (int i = 0; i < 7; i++)
```

```
            {
```

```
                Console.Write(string.Format("Enter number {0} > ", i + 1));
```

```
                playerOneEntries[i] = (Convert.ToInt16(Console.ReadLine()));
```

```
            }
```

```
            Console.Clear();
```

```
            for (int i = 0; i < 6; i++)
```

```
            {
```

```
                Console.Write(string.Format("{0} value is {1} ", (i > 0 ? "the next" :
```

```
                "Player 2 the first"), playerOneEntries[i]));
```

```
                Console.WriteLine();
```

```
                Console.Write("Higher (H) or lower (L) > ");
```

```
                result = Console.ReadKey().Key.ToString().ToUpper();
```

```
                Console.WriteLine();
```

```
                if (result == "H")
```

```
                {
```

```
                    if (playerOneEntries[i + 1] > playerOneEntries[i])
```

```
                    {
```

```
                        Console.Write("Correct, ");
```

```
                    }
```

```
                else
```

```
                {
```

```
                    playerTwoFailsOnce = true;
```

```
                    Console.Write("Incorrect, ");
```

```
                }
```

```
            }
```

```
            else if (result == "L")
```

```
            {
```

```
                if (playerOneEntries[i + 1] < playerOneEntries[i])
```

```
                {
```

```
                    Console.Write("Correct, ");
```

```
                }
```

```
            else
```

Validated?

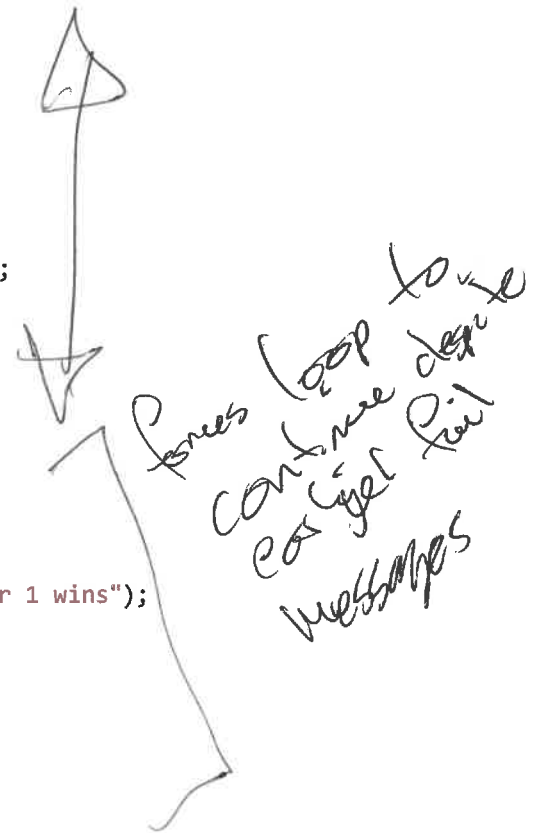
class!

good, but overly complex. Validated?

```
        {
            playerTwoFailsOnce = true;
            Console.WriteLine("Incorrect, ");
        }
    }
    else
    {
        Console.WriteLine("Invalid answer..");
    }
}

Console.WriteLine();
Console.WriteLine();

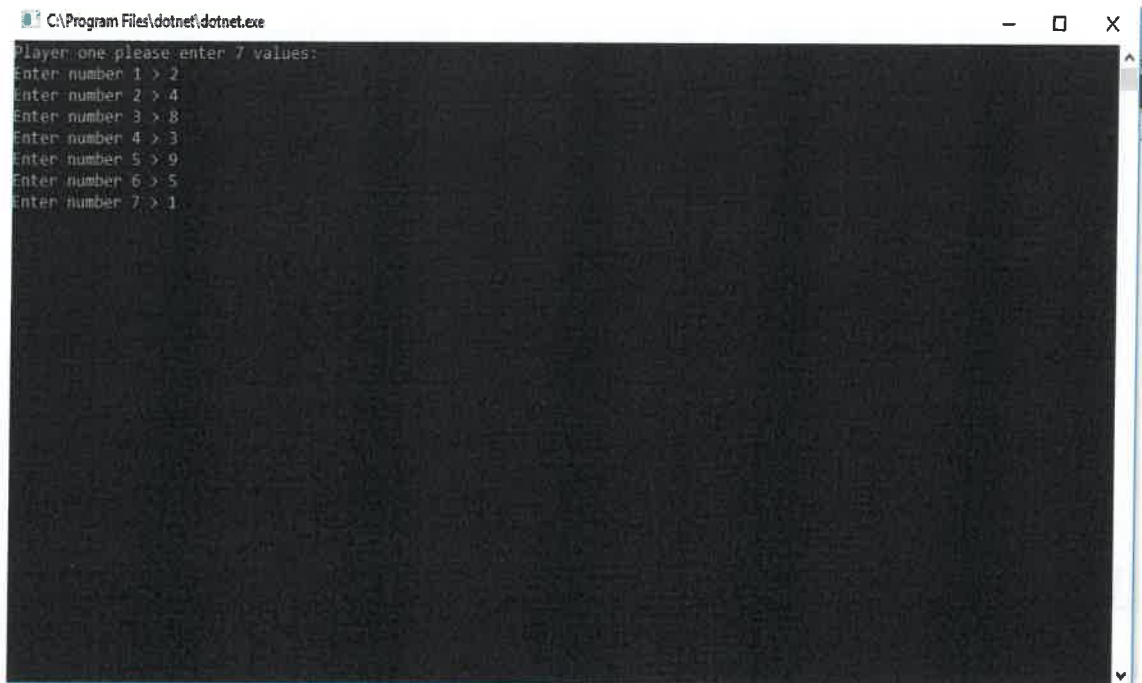
if (playerTwoFailsOnce == true)
{
    Console.WriteLine("Incorrect, sorry player 1 wins");
}
else
{
    Console.WriteLine("Player 2 wins");
}
Console.Read();
}
}
```



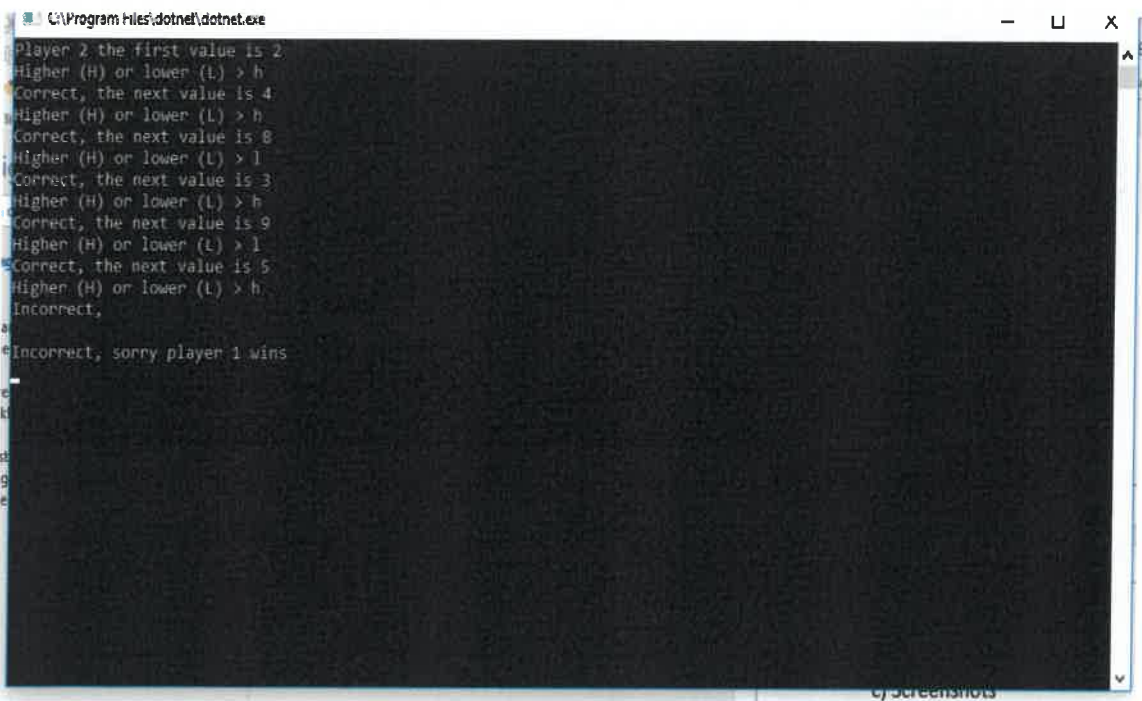
b) List of problems

1. Display menu
2. Ask user to enter 7 values
3. Get user input
4. Validate if it's a positive number and above 0
5. Ask user to enter H for higher value and L for lower value
6. Validate – it should not be a number or any other letters

c) Screenshots



```
C:\Program Files\dotnet\dotnet.exe
Player one please enter 7 values:
Enter number 1 > 2
Enter number 2 > 4
Enter number 3 > 8
Enter number 4 > 3
Enter number 5 > 9
Enter number 6 > 5
Enter number 7 > 1
```



```
C:\Program Files\dotnet\dotnet.exe
Player 2 the first value is 2
Higher (H) or lower (L) > h
Correct, the next value is 4
Higher (H) or lower (L) > h
Correct, the next value is 8
Higher (H) or lower (L) > l
Correct, the next value is 3
Higher (H) or lower (L) > h
Correct, the next value is 9
Higher (H) or lower (L) > l
Correct, the next value is 5
Higher (H) or lower (L) > h
Incorrect,
Incorrect, sorry player 1 wins
c:/screenshots
```

d) Description of tests

First of all, the player 1 needs to enter 7 values. Player 2 cannot see that numbers. After entering the 7 values, the screen will clear and the first value will be shown to player 2. Player 2 needs to guess if the second value is higher or lower than the first value. In my case, the first value is 2. The second value is 4. So player 2 guessed correctly higher. Then it will ask the next value if higher or lower until player 2 enter incorrect guess.

2) Exercise 2

a) Coding

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
//REKA RAGUPATHY
namespace HiddenTreasures
{
```

```
    class Program
    {
```

```
        static string[,] game = {
```

```
{ " ", " ", " ", " ", " ", " ", " ", " " },
{ " ", " ", " ", " ", " ", " ", " ", " " },
{ " ", " ", " ", " ", " ", " ", " ", " " },
{ " ", " ", " ", " ", " ", " ", " ", " " },
{ " ", " ", " ", " ", " ", " ", " ", " " },
{ " ", " ", " ", " ", " ", " ", " ", " " },
{ " ", " ", " ", " ", " ", " ", " ", " " },
{ " ", " ", " ", " ", " ", " ", " ", " " }
```

```
};
```

```
static void Main(string[] args)
{
```

```
    int[] playerOneEntries = new int[6];
```

```
    int[] playerTwoEntries = new int[6];
```

```
    string result = "";
```

```
    bool playerTwoFailsOnce = false;
```

```
    string columnPosition;
```

```
    string rowPosition;
```

```
    string index;
```

```
Console.WriteLine("Player 1 place treasures now: ");
```

```
for (int i = 0; i < 1; i++)
{
```

```
    Console.Write(string.Format("X: "));
```

```
    playerOneEntries[i] = (Convert.ToInt16(Console.ReadLine()));
```

```
    Console.Write(string.Format("Y: "));
```

```
    playerOneEntries[i] = (Convert.ToInt16(Console.ReadLine()));
```

```
    Console.WriteLine("Treasure 1 placed");
```

```
    Console.Write(string.Format("X: "));
```

```
    playerOneEntries[i] = (Convert.ToInt16(Console.ReadLine()));
```

```
    Console.Write(string.Format("Y: "));
```

```
    playerOneEntries[i] = (Convert.ToInt16(Console.ReadLine()));
```

```
    Console.WriteLine("Treasure 2 placed");
```

```
    Console.Write(string.Format("X: "));
```

```
    playerOneEntries[i] = (Convert.ToInt16(Console.ReadLine()));
```

```
    Console.Write(string.Format("Y: "));
```

```
    playerOneEntries[i] = (Convert.ToInt16(Console.ReadLine()));
```

```
    Console.WriteLine("Treasure 3 placed");
```

```
}
```

over-writing data

c) Screenshots

