

**APPENDIX GA36a**

LEVEL 4 ASSESSMENT SPECIFICATION

Student name:	Sean Samuel	Student ID number:	SAM18841586
Programme:	Computing Portfolio		
Module:	Software Development		
Module code:	ACCA4017	Contribution to Overall Module Assessment (%):	70%
Lecturer:	Geoff Dawkins	Internal Verifier:	Paul John
Assignment Title:	Programming in C	Word count (or equivalent):	2800
Submission deadline:	2/4/19	Return date of provisional marks & written feedback:	30/4/19
Submission method:	All written assessments, where practical and possible, must be submitted via Turnitin unless otherwise instructed by the Lecturer. (Please DO NOT put this assessment specification into Turnitin or it will match many similarities with other students' submissions.) Alternative submission method (if applicable): Late submission of the assessment will result in a late penalty mark. Penalties for late submission: Up to one week late, maximum mark of 40%. Over one week late, 0%. Only the Extenuating Circumstances Panel may approve a change to submission dates.		
Academic honesty / referencing:	Academic honesty is required. In the main body of your submission you must give credit to authors on whose research and ideas your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have used, cited or quoted in order to complete this assessment.		

Contents

Program 1	3
Problems before the start of coding	3
Problems encountered while coding	3
Problem 1.....	3
Problem 2.....	5
Problem 3.....	6
Program 2.....	7
Problems before the start of coding	7
Appendix	8
Program 1 Code.....	8
Program 2 Code.....	10

Program 1

Problems before the start of coding

1. Display a menu that asks the first player to enter 7 values
2. Create an array that will hold the users input
3. Validation so the user can only enter a number
4. Display error message if they enter something other than a number and notify them to enter a number
5. Validation so that the numbers can only be between 1 and 10
6. Display error message if they enter a number outside the range of 1-10 and notify them to enter a number within that range
7. Show to the second player the first value that player one entered
8. Ask player two whether the first value is higher or lower
9. Show player two that they should enter an L for lower and a H for higher
10. Validation so player two can only enter either an L or a H and no numbers
11. Display error message if they enter an incorrect value
12. If player two is correct with their guess, then show that they were correct and proceed to the second value
13. Carry on until player two wins in a loop if they keep guessing correctly
14. Create an IF statement within the loop to see if the player keeps guessing correctly and if not then end the game
15. End the game and show that player one wins if they enter an incorrect guess

Problems encountered while coding

Problem 1

Had a problem with creating the ints for the user to input the 7 values.

```
int numbers = int.Parse(Console.ReadLine());
```

I first tried to create one int to hold the numbers. Then it took me a while to realise I had to create 7 individual ints for the user to input.

```
int numberOne = int.Parse(Console.ReadLine());  
int numberTwo = int.Parse(Console.ReadLine());  
int numberThree = int.Parse(Console.ReadLine());  
int numberFour = int.Parse(Console.ReadLine());  
int numberFive = int.Parse(Console.ReadLine());  
int numberSix = int.Parse(Console.ReadLine());  
int numberSeven = int.Parse(Console.ReadLine());
```

Problem 2

I then had a problem with the loops, where it was running the incorrect amount of times.

```
for (var numCount = 0; numCount < 7; numCount++)
```

I realised I had to change the number to 6 as that was the last number in the array, rather than 7 for 7 values.

```
for (var numCount = 0; numCount < 6; numCount++)
```

Test of it working:

```
Hello, we're going to play a game. What is your name?
Sean
Hello Sean, you are player one. Please enter 7 values
3
4
5
6
4
3
5

Player Two, what is your name?
Alex
Hello Alex
Please guess if the next number is higher or lower than 3
H for higher and L for lower
h
You are correct!
Please guess if the next number is higher or lower than 4
H for higher and L for lower
h
You are correct!
Please guess if the next number is higher or lower than 5
H for higher and L for lower
h
You are correct!
Please guess if the next number is higher or lower than 6
H for higher and L for lower
l
You are correct!
Please guess if the next number is higher or lower than 4
H for higher and L for lower
l
You are correct!
Please guess if the next number is higher or lower than 3
H for higher and L for lower
h
Congratulations Alex, you have won the game
```

Problem 3

I had a problem with the validation to see if the user enters a value outside of the range 1-10 and if the user enters something other than a number but I could not get it working.

```
string userInput;
int userNumber;
int userOutput;

//Validation to check if the user enters something other than a number
if (int.TryParse(userInput, out userNumber))
{
    userOutput = userNumber;
}
else
{
    Console.WriteLine("Value entered was not a number. Please enter a number between 1 and 10");
}

//Validation to check if the user enters a number outside the range of 1-10
if (userNumber < 1 || userNumber > 10)
{
    Console.WriteLine("Please enter a number inbetween 1 and 10");
}
```

Program 2

Problems before the start of coding

1. Create menu prompting player one to place 3 treasures
2. Validation so that player one can only enter values between 0-7 on the 8x8 grid
3. Display error message if they do, telling them to enter a value between 0-7
4. Validation so that player one cannot enter anything other than a number
5. Display error message if they do, telling them to enter a number
6. Create array to hold player one's values
7. Show the grid to player 2 and ask them to guess where the values are hidden
8. Validation so that player two can only enter values between 0-7 on the 8x8 grid
9. Display error message if they do, telling them to enter a value between 0-7
10. Validation so that player two cannot enter anything other than a number
11. Display error message if they do, telling them to enter a number
12. After they enter their first guess, show the grid again with either an N for a miss or an X for a hit
13. Repeat until they either win or lose
14. Show player two that they have 3 lives and it reduces each time they guess wrong
15. If player two guesses all 3 correct, then show them that they have won the game
16. If player two fails to guess all 3 then show that player one wins

Appendix

Program 1 Code

```
//SAM18841586
```

```
//Make the program more user friendly by asking for names
Console.WriteLine("Hello, we're going to play a game. What is your name?\n");
string nameOne = Console.ReadLine();

Console.WriteLine("Hello {0}, you are player one. Please enter 7 values", nameOne);
```

```
//Create ints for each number to use in the array - The user will input these
int numberOne = int.Parse(Console.ReadLine());
int numberTwo = int.Parse(Console.ReadLine());
int numberThree = int.Parse(Console.ReadLine());
int numberFour = int.Parse(Console.ReadLine());
int numberFive = int.Parse(Console.ReadLine());
int numberSix = int.Parse(Console.ReadLine());
int numberSeven = int.Parse(Console.ReadLine());
```

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

```
//Array for numbers
numbers[0] = numberOne;
numbers[1] = numberTwo;
numbers[2] = numberThree;
numbers[3] = numberFour;
numbers[4] = numberFive;
numbers[5] = numberSix;
numbers[6] = numberSeven;
```

```
/*string userInput;
int userNumber;
int userOutput;
```

```
Validation to check if the user enters something other than a number
if (int.TryParse(userInput, out userNumber))
```

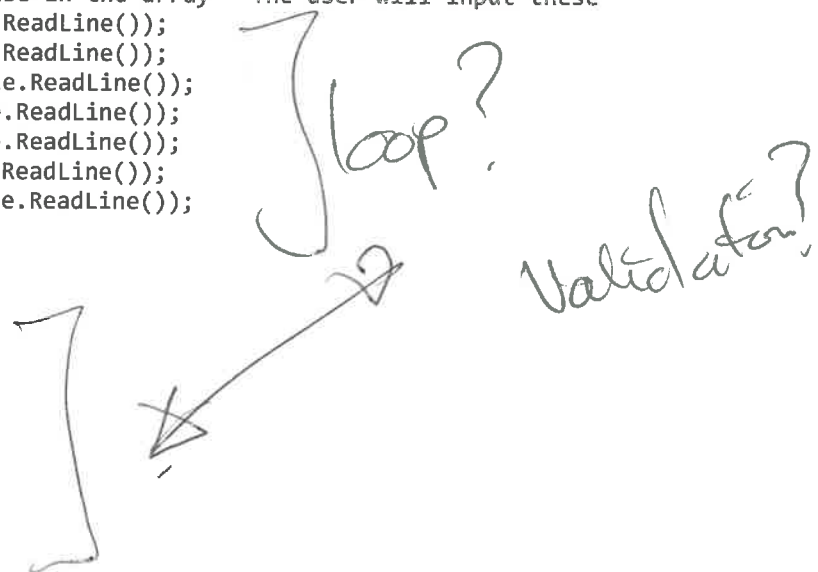
```
{
    userOutput = userNumber;
}
else
{
```

```
    Console.WriteLine("Value entered was not a number. Please enter a number between 1
and 10");
}
```

```
//Validation to check if the user enters a number outside the range of 1-10
if (userNumber < 1 || userNumber > 10)
```

```
{
    Console.WriteLine("Please enter a number inbetween 1 and 10");
}
*/
```

```
Console.Write("\n");
```



```

Console.WriteLine("Player Two, what is your name?\n");
string nameTwo = Console.ReadLine();

Console.Write("\n");

Console.WriteLine("Hello {0}", nameTwo);

//Loop that will keep repeating if the user guesses correctly and ends if they lose or
win
for (var numCount = 0; numCount < 6; numCount++)
{
    //The {1} is for the variable numbers and counts each number while going around the
loop
    Console.WriteLine("Please guess if the next number is higher or lower than {1}",
nameTwo, numbers[numCount].ToString());
    Console.WriteLine("H for higher and L for lower\n");

    string highOrLow = Console.ReadLine().ToLower();

    //If to check if they entered H for higher
    if (highOrLow == "h")
    {
        if (numbers[numCount] < numbers[numCount + 1])
        {
            if (numCount == 5)
            {
                Console.WriteLine("Congratulations {0}, you have won the game",
nameTwo);
                Console.ReadLine();
            }
            else
            {
                Console.WriteLine("You are correct!");
            }
        }
        else
        {
            Console.WriteLine("Sorry, you are incorrect. {0} wins", nameOne);
            Console.ReadLine();
            break;
        }
    }

    // If they didn't enter H then run this code
    else
    {
        Console.Write("\n");

        if (numbers[numCount] > numbers[numCount + 1])
        {
            if (numCount == 5)
            {
                Console.WriteLine("Congratulations {0}, you have won the game",
nameTwo);
                Console.ReadLine();
            }
            else
            {
                Console.WriteLine("You are correct!");
            }
        }
        else
        {
            Console.WriteLine("Sorry, you are incorrect. {0} wins", nameOne);

```

```
        Console.ReadLine();  
        break;  
    }  
}  
  
Console.ReadLine();
```

Program 2 Code

```
static void Main(string[] args)  
{  
    Console.WriteLine("Hello, we're going to play a game. What is your name?");  
    string nameOne = Console.ReadLine();  
  
    Console.WriteLine("Hello {0}, you are player one. You must place treasures and player 2  
must find them", nameOne);  
    Console.WriteLine("Player one, please place your treasures now");  
  
    Console.WriteLine("\n");  
  
    int xValue;  
    int yValue;  
    int xValue2;  
    int yValue2;  
    int xValue3;  
    int yValue3;  
    bool win = false;  
  
    while (!win)  
    {  
        Console.Write("X:");  
        xValue = int.Parse(Console.ReadLine());  
        Console.Write("Y:");  
        yValue = int.Parse(Console.ReadLine());  
  
        cordValues[xValue, yValue] = "";  
  
        Console.Write("X:");  
        xValue2 = int.Parse(Console.ReadLine());  
        Console.Write("Y:");  
        yValue2 = int.Parse(Console.ReadLine());  
  
        cordValues[xValue2, yValue2] = "";  
  
        Console.Write("X:");  
        xValue3 = int.Parse(Console.ReadLine());  
        Console.Write("Y:");  
        yValue3 = int.Parse(Console.ReadLine());  
  
        cordValues[xValue3, yValue3] = "";  
  
        ShowCordValues();  
  
        if (!win)  
        {  
            Console.Write("X:");  
            xValue = int.Parse(Console.ReadLine());  
            Console.Write("Y:");
```

```

yValue = int.Parse(Console.ReadLine());

cordValues1[xValue, yValue] = "X";

if (cordValues == cordValues1)
{
    Console.WriteLine("You have found a treasure");
}

Console.Write("X:");
xValue2 = int.Parse(Console.ReadLine());
Console.Write("Y:");
yValue2 = int.Parse(Console.ReadLine());

cordValues1[xValue2, yValue2] = "X";

if (cordValues == cordValues1)
{
    Console.WriteLine("You have found a treasure");
}

Console.Write("X:");
xValue3 = int.Parse(Console.ReadLine());
Console.Write("Y:");
yValue3 = int.Parse(Console.ReadLine());

cordValues1[xValue3, yValue3] = "X";

if (cordValues == cordValues1)
{
    Console.WriteLine("You have found a treasure");
}

ShowCordValues();
}

```

} does not work

} does not work.

```

static string[,] cordValues =
{
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" }
};

static string[,] cordValues1 =
{
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" },
    { "", "", "", "", "", "", "" }
};

public static void ShowCordValues()
{
    Console.WriteLine("");
}

```

```
    for (int i = 0; i < 7; i++)
    {
        Console.WriteLine(" {0} | {1} | {2} | {3} | {4} | {5} | {6} | {7} ", cordValues[i,
0], cordValues[i, 1], cordValues[i, 2], cordValues[i, 3], cordValues[i, 4], cordValues[i, 5],
cordValues[i, 6], cordValues[1,7]);

    }
    Console.WriteLine("");
}
```