

Jordan Griffiths 1805091

by Jordan Griffiths

FILE	11995_JORDAN_GRIFFITHS_JORDAN_GRIFFITHS_1805091_178138_425889402.DOCX (49.9K)		
TIME SUBMITTED	05-APR-2019 09:45PM (UTC+0100)	WORD COUNT	522
SUBMISSION ID	104244072	CHARACTER COUNT	2436

Assignment 1 Programming

Jordan Griffiths 1805091

Contents:

Page 1: Introduction for the program.

Page 2: Flowchart of the program.

Page 3: Testing.

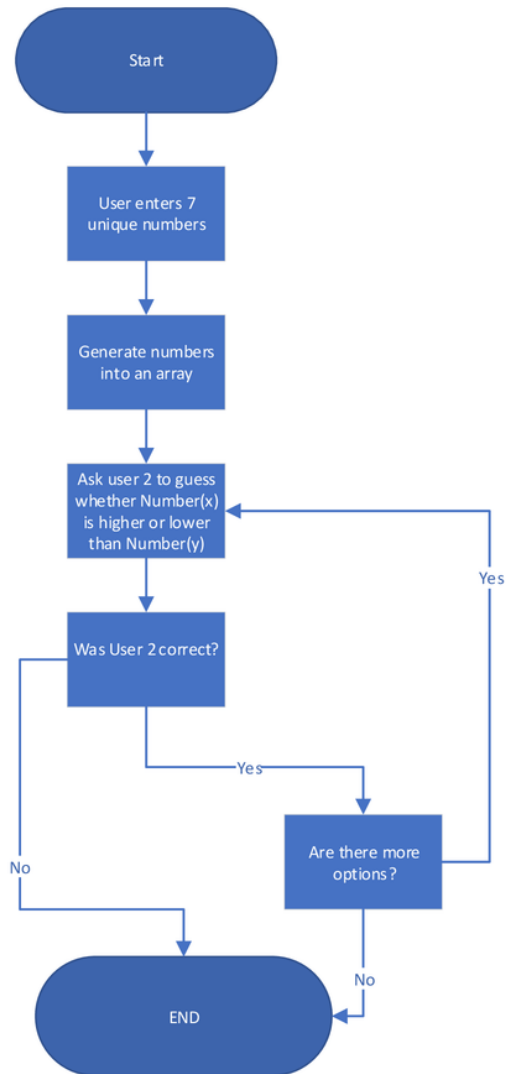
Page 3: User guide.

Page 4: Example of code.

Introduction:

The purpose of this report is to document the numbers guessing game that was asked for during this assignment. The requirement for the code was to create a program that was able to take 7 numbers that the user inputs and then for a second user to try and guess whether the next number is higher or lower than the first number that is shown, the code will keep repeating if user 2 is correct until they have guessed all correctly, if the user gets one answer incorrect the game will end.

Flowchart of information:



Testing

Test No	Test	Input	Expected Result	Actual Result
1	Inputting all numbers as 1-7	1, 2, 3, 4, 5, 6, 7	Each answer is Higher	Higher answer is correct each time.
2	Inputting all numbers as 7-1	7, 6, 5, 4, 3, 2, 1	Each answer is Lower	Lower answer is correct each time
3	Does the game end after a single incorrect answer	An incorrect answer	Game ends	Game ends showing incorrect screen
4	How does the game react to duplicate numbers input	Duplicate numbers into the input	Game error	Game ends showing incorrect screen.



User guide.

To correctly use this program User one must enter 7 unique numbers into the system, then user two must guess whether the number shown is either Higher or Lower than the first number shown on the screen by inputting the number 1 for Higher or the number 2 for Lower. Once player 2 presses enter the game will give you the correct answer, if player 2 was correct the next question will be asked however if player 2 is incorrect then the game will end with player 1 winning the game.

```

#include "pch.h"
#include <cstdlib>
#include <time.h>
#include <iostream>

using namespace std;

int main() {
    int number[7];

    cout << "Player 1, please enter your 7 numbers!" << endl;

    cin >> number[0];
    cin >> number[1];
    cin >> number[2];
    cin >> number[3];
    cin >> number[4]; 
    cin >> number[5];
    cin >> number[6];
    int i;
    int guess;
    int correct;

    do {
        for (i = 0; i < 6; i++) 
        {
            cout << "Player 2, this number is: " << number[i] << ", is the
next number higher or lower? (Type 1 for Higher or 2 for Lower)" << endl;

            cin >> guess;

            if (guess == 1)
            {
                if (number[i] > number[i + 1])
                {
                    cout << guess << endl;
                    correct = 1;
                    cout << "Correct!" << endl;
                }

                else if (number[i] < number[i+1])
                {
                    correct = 2;
                    cout << "Incorrect, player 1 wins" << endl;
                    break;
                }
            }

            else if (guess == 2)
            {
                if (number[i] < number[i + 1])
                {
                    correct = 1;
                    cout << guess << endl;
                    cout << "Correct!" << endl;
                }
                else if (number[i] > number[i + 1])
                {
                    correct = 2;

```

```
        cout << guess << endl;
        cout << "Incorrect, player 1 wins" << endl;
        break;
    }
}
} while (correct == 1);
return 0;
}
```

FINAL GRADE

32/100

GENERAL COMMENTS

Instructor

Exercise 1 (50) 32

Exercise 2 (30) 0

Exercise 3 (20) 0

Generally the code of exercise 1 is a bit messy. I wanted to see more evidence of problem solving and sub-division of problems with associated functions.

PAGE 1

PAGE 2

PAGE 3



Comment 1

What about player 2 winning?

PAGE 4



Comment 2

Use a loop for this!!

Put in a function



Comment 3

You are solving multiple problems all in the main, we want to avoid doing this!!

Make life easier for yourself by breaking the code down into small functions.

PAGE 5



Comment 4

How does player 2 win?