

Shamuel Hashan 1803153

by Samuel Hashan

FILE	11987_SHAMUEL_HASHAN_SHAMUEL_HASHAN_1803153_178138_155782784.DOCX (911.38K)		
TIME SUBMITTED	05-APR-2019 11:19AM (UTC+0100)	WORD COUNT	3339
SUBMISSION ID	104171455	CHARACTER COUNT	15082

Programming in C

Software Development

University of Wales Trinity St David

Web Development

Lecturer: Mike Dacey

Shamuel Hashan

April 5 2019

1803153

Contents

- Introduction
- Part A
 - Planning
 - Developing code
 - Problems
- Conclusion
- Appendix

Introduction

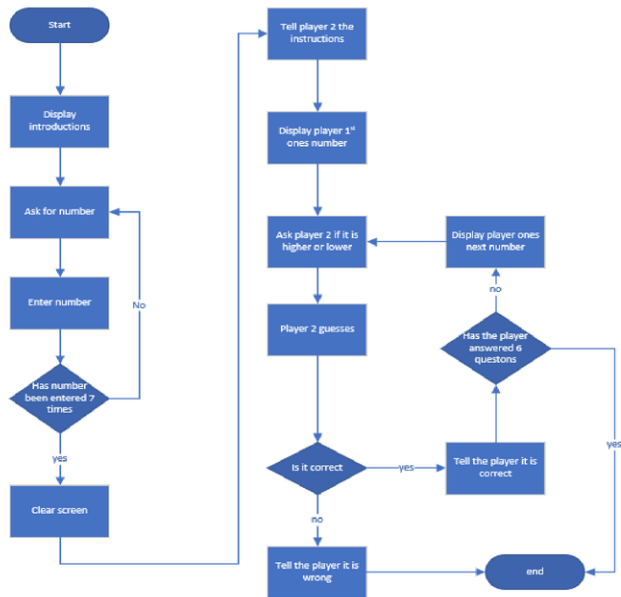
For this assignment I will be looking at how to programme using c++. I am going to make a game using this programming language. I will be documenting how I programmed the game. I will look at developing the code, testing plans and any issues I had. I will list these issues from easiest to hardest to fix.

The game

I created a game with two players. Player 1 will need enter 7 numbers and player 2 will have to guess if the numbers are higher or lower. If player 2 guesses if all the numbers are higher or lower to what player 1 has entered. If it is wrong, then player 2 will lose but if player 2 gets it all right then it's a win.

Planning

This is a flow chart I created when I was planning how to do my code and what it should achieve. I decided that I would need to use a loop for player ones input on entering the numbers and for player 2s input on guessing if they are higher or lower. It shows what the game should do and how it gets to the end.



Testing and code

Test 1

This is the 1st test I did. The aim of this test is to see if the instructions for player 1 is displayed when the game is started. It should display the instructions. This is the code that I entered.

```
cout << "Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower." << endl;
```

I tested the code by starting the game. The code worked since the instructions was displayed for player one.

Test 2

The goal of the 2nd test I did was to see if the code that is supposed to display how player 2 is going to play the game. It is supposed to display under the instructions for player 1.

```
cout << "Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy" << endl;
```

I tested it by playing the game and found that my code worked.

This is the instructions being shown in the game.

```
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.  
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy  
please enter a number between 1 and 10
```

Test 3

This test was to check if the code for player one's input worked. It should display a question asking player 1 to enter a number. It should display *"please enter a number between 1 and 10,"* This is the code I used that does this.

```
cout << "please enter a number between 1 and 10" << endl;
```

It seems like it worked as it did display *"please enter a number between 1 and 10."*

```
please enter a number between 1 and 10
```

Test 4

The purpose of the 4th test is to see if the code for the player 1 input works. After player one is prompted to, player one should be able to enter a number. I tested this by entering a number. This is the code that I used to make it work.

```
int numbers[7];
    for (count = 0; count < 7; count++) {
        cout << "please enter a number between 1 and 10" << endl;
        getline(cin, input);
        stream.str(input);
        stream >> numbers[count];
        stream.clear();
    }
```

To test this, I entered 3 to see if it would allow me to input a number, It should allow me to input 3 and it did.

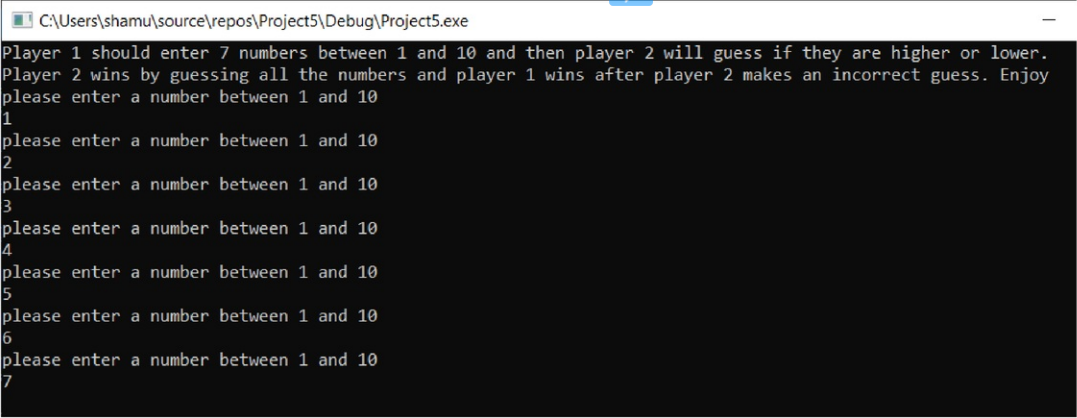
```
please enter a number between 1 and 10
3
```

Test 5

This test is for the loop. It should repeat the previous two tests till player one has entered a number 7 times. It should then repeat this till the user has entered 7 numbers. I used a for loop for this as you can see in the code below.

```
int numbers[7];
for (count = 0; count < 7; count++) {
    cout << "please enter a number between 1 and 10" << endl;
    getline(cin, input);
    stream.str(input);
    stream >> numbers[count];
    stream.clear();
}
```

I tested the loop by playing the game and entering a number 7 times, I entered 1, then 2, then 3, then 4, then 5, then 6 and then 7. I think the code worked since it allowed me to enter a number 7 times.



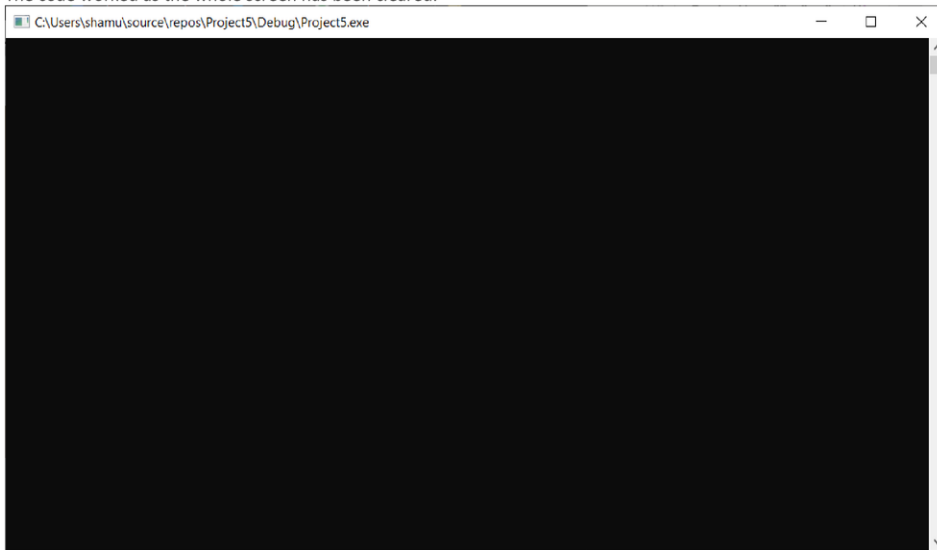
```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy
please enter a number between 1 and 10
1
please enter a number between 1 and 10
2
please enter a number between 1 and 10
3
please enter a number between 1 and 10
4
please enter a number between 1 and 10
5
please enter a number between 1 and 10
6
please enter a number between 1 and 10
7
```

Test 6

This test is to see everything in the screen is cleared after player one entered 7 numbers. I tested this by pressing enter after the 7th number. The screen should be cleared. This is the code I used for this.

```
system("CLS"); // clears the screen
```

The code worked as the whole screen has been cleared.



TEST 7

This tests if player one's numbers are recorded. It should display what player one's number was and ask player two to guess if it is higher or lower. The code here is supposed to display what player one has picked

```
string guess(int numbers[],int number)
{
    string input;
    cout << "The number player one picked was " << numbers[number] << " Is the next number higher or lower?" << endl;
    getline(cin, input);
    return input;
}
```

I picked 4 as the 1st number for player one and this is it showing what player ones number is

```
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.
The number player one picked was 4 Is the next number higher or lower?
```

The 1st number I entered was 4 so this does prove that this works.

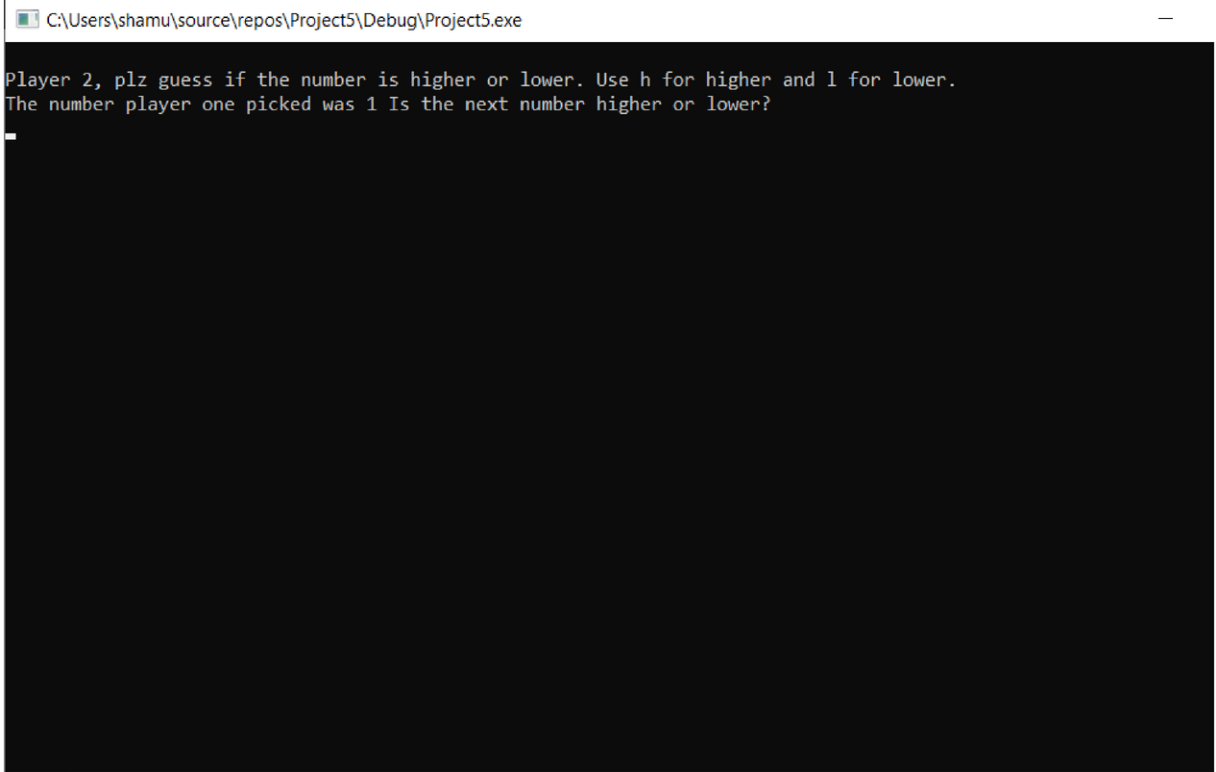
Test 8

This test was to display if the code for player two's instructions worked. It should say "player 2 plz guess if the number is higher or lower, *use h for higher and l for lower*," This is the code I used to display this. It worked as I expected since it did display some text saying "Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower."

```
cin.get();  
cout << "Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower. " << endl;  
guess(numbers);  
HorL(numbers);  
  
return(0);
```

C:\Users\shamu\source\repos\Project5\Debug\Project5.exe

```
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.  
The number player one picked was 1 Is the next number higher or lower?  
-
```



```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe

Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.
The number player one picked was 1 Is the next number higher or lower?
_
```

Test 9

The next test was to see if the code for displaying the text letting player 2 know what the previous number was worked. To test it, I entered 1 for the 1st number as player one. This means that it should say, *"the number player one picked was 1. Is the next number higher or lower?"* This is the code I used.

```
string guess(int numbers[],int number)
{
    string input;
    cout << "The number player one picked was " << numbers[number] << ". Is the next number higher or lower?" << endl;
    getline(cin, input);
    return input;
}
```

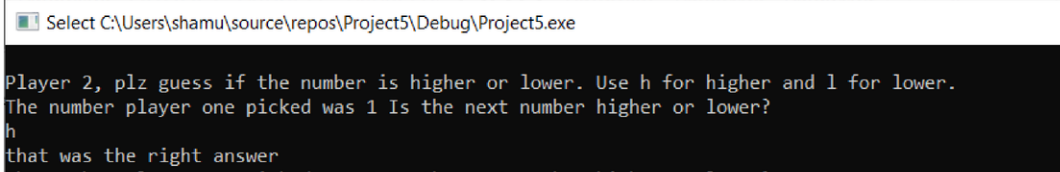
```
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.
The number player one picked was 1 Is the next number higher or lower?
```

The code worked since it did display *"the number player one picked was 1. Is the next number higher or lower?"*

Test 10

The next thing I had to test was to see if the code for checking if the game if the input player 2 gave was correct for when they enter "h" when the next number is higher. To test this, I made the second number for player one, 2. This number is higher than the first number I entered which means that if the user enters, h then it should say *"that was the right answer."* I entered h and it did display. *"that was the right answer."*

```
if (higherlower == "h")
{
    if (numbers[count] < numbers[count + 1])
    {
        cout << "that was the right answer " << endl;
    }
}
```



```
Select C:\Users\shamu\source\repos\Project5\Debug\Project5.exe

Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.
The number player one picked was 1 Is the next number higher or lower?
h
that was the right answer
```

Test 11

I then tested to see if the game recognised when that player 2 is wrong when they don't enter "h" when the next number is higher. I tested this out making the second number for player one, 2. This number is higher than the first number I entered which means that if the user enters, I then it should say "that was the wrong answer." I entered l and it did display. "that was the wrong answer." This means that the code worked well.

C:\Users\shamu\source\repos\Project5\Debug\Project5.exe

```
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.  
The number player one picked was 1. Is the next number higher or lower?  
l  
that was the wrong answer
```

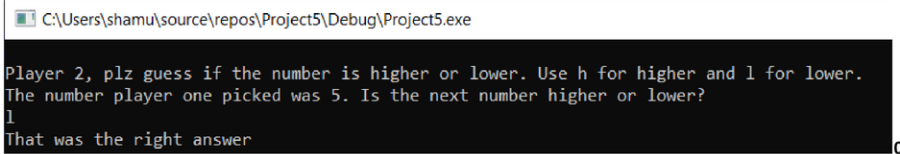
This is the code I entered to make this work.

```
for (count = 0; count < 6; count++) {  
    higherlower = guess(numbers, count);  
    if (higherlower == "h")  
    {  
        if (numbers[count] < numbers[count + 1])  
        {  
            cout << "that was the right answer " << endl;  
        }  
    }  
    else  
    {  
        cout << "that was the wrong answer" << endl;  
        cin.get();  
        break;  
    }  
}
```

Test 12

This test is for player two's input when the next number is lower. The purpose of this test was to see if the game recognised when that the user is right when the next number is lower, and they enter l. To test this, I entered 5 as the 1st number and I made the number after that 3. If the user enters l then it should say that *"That was the right answer."* After I entered l, the game did display say *"that was the right answer"* which means that the test was successful. This is the code I used for this.

```
else
{
    if (numbers[count] >= numbers[count + 1])
    {
        cout << "That was the right answer " << endl;
    }
}
```



```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.
The number player one picked was 5. Is the next number higher or lower?
l
That was the right answer
```

Test 13

This test is for player two's input when the next number is lower. The purpose of this test was to see if the game recognised when that the user is wrong when the next number is lower, but they enter otherwise. To test this, I entered 5 as the 1st number and I made the number after that 3. If the user enters h, then it should say that *"That was the wrong answer."* After I entered h, the game did display say *"that was the wrong answer"* which means that the test was successful.

```
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.  
The number player one picked was 5. Is the next number higher or lower?  
h  
that was the wrong answer
```

Test 14

The purpose of this test is to see if a loop works with the previous tests from all the tests after test 10. This should allow player 2 to guess if the numbers that player one has picked. It should repeat the process at about 6 times if the user gets it right all the time. I tested this bit of code by entering 1 as the 1st number, 2 as the 2nd number, 3 as the 3rd number, 4 as the 4th number, 5 as the 5th number, 6 as the 6th number and 7 as the 7th and final number. As player 2, I entered h for all the answers. It should say *"that was the right answer"* for all the answers. After I did that, it did let me enter the answer 6 times after I kept answering with h which means that the code must have worked, and the test was successful.

This is the code.

```
for (count = 0; count < 6; count++) {
    higherlower = guess(numbers, count);
    if (higherlower == "h")
    {
        if (numbers[count] < numbers[count + 1])
        {
            cout << "that was the right answer " << endl;
        }
        else
        {
            cout << "that was the wrong answer" << endl;
            cin.get();
            break;
        }
    }
    else
    {
        if (numbers[count] >= numbers[count + 1])
        {
            cout << "That was the right answer " << endl;
        }
        else
        {
            cout << "that was the wrong answer " << endl;
            cin.get();
            break;
        }
    }
}
```

```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.
The number player one picked was 1. Is the next number higher or lower?
h
that was the right answer
The number player one picked was 2. Is the next number higher or lower?
h
that was the right answer
The number player one picked was 3. Is the next number higher or lower?
h
that was the right answer
The number player one picked was 4. Is the next number higher or lower?
h
that was the right answer
The number player one picked was 5. Is the next number higher or lower?
h
that was the right answer
The number player one picked was 6. Is the next number higher or lower?
h
```

Test 15

This is to test the code for the end the game when the player gets it wrong. The purpose of this test is to see if the game ends after the user gets an answer wrong. To test this, I input 7 as the 1st number for player one. For the 2nd number I input 4. This is lower than the 1st number I entered. For player 2, I am going input h as the 1st answer. This is the wrong answer as "l" would have been the correct answer. The game should display *"that was the wrong answer"* and then exit the game.

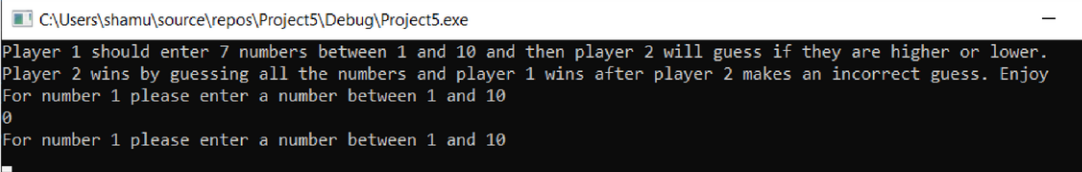
```
    {  
        cout << "that was the wrong answer " << endl;  
        cin.get();  
        break;  
    }  
}
```

Test 16

This is a test for the validation of player one's input. The purpose of this test is to see if the programme does not let player advance unless they use a number between 1 and 10. To test this I entered 0 as the 1st number. I figured I would need to use a while loop to make it work. To test this I started the program and entered 0 to see what happens if the number is below 1. It should say *"for number 1 please enter a number between 1 and 10."* Look at the 3rd in the list for problems for more on this.

```
while (numbers[count] < 1 || numbers[count] > 10)  
{  
    cout << "For number " << count << " please enter a number between 1 and 10" << endl;  
    getline(cin, input);  
    stream.str(input);  
    stream >> numbers[count];  
    stream.clear();  
}
```

The code worked. It still says *"for number 1 please enter a number between 1 and 10"* which is what it should do.



```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe  
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.  
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy  
For number 1 please enter a number between 1 and 10  
0  
For number 1 please enter a number between 1 and 10
```

Test 17

The purpose of this test is to see what happens when player one's input is a number above 10. To test this, I made player 1 input 11. It should display *"for number 1 please enter a number between 1 and 10"* since 11 does not fit between those two numbers. I used the same code as test 16. The code worked as it did display the same line again.

```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy
For number 1 please enter a number between 1 and 10
11
For number 1 please enter a number between 1 and 10
```

Test 18

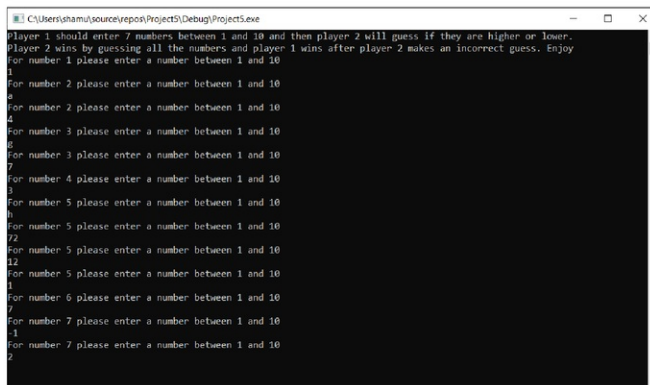
I tested the validation again. This test is to see what happens when player one enters something that is not a number. It should display *"for number 1 please enter a number between 1 and 10."* It did say that which means the code works for letters.

```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy
For number 1 please enter a number between 1 and 10
11
For number 1 please enter a number between 1 and 10
a
For number 1 please enter a number between 1 and 10
```

Test 19

I then tested the loop again, This was to check if the validation worked in a loop. I entered 1 as the 1st number and then a as the 2nd. The 1st input should be fine, but for the 2nd number it should display *"For number 2 please enter a number between 1 and 10."* It did work as I expected which means the code worked well. I entered random numbers for the rest of the loop to see if they also worked and they did. This is the code of the while statement in a loop.

```
int numbers[7];
for (count = 1; count < 8; count++)
{
    while (numbers[count] < 1 || numbers[count] > 10)
    {
        cout << "For number " << count << " please enter a number between 1 and 10" << endl;
        getline(cin, input);
        stream.str(input);
        stream >> numbers[count];
        stream.clear();
    }
}
```



```
C:\Users\shamuel\source\repos\Project5\Debug\Project5.exe
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy
For number 1 please enter a number between 1 and 10
1
For number 2 please enter a number between 1 and 10
a
For number 3 please enter a number between 1 and 10
2
For number 3 please enter a number between 1 and 10
8
For number 3 please enter a number between 1 and 10
7
For number 4 please enter a number between 1 and 10
3
For number 5 please enter a number between 1 and 10
0
For number 5 please enter a number between 1 and 10
22
For number 5 please enter a number between 1 and 10
12
For number 5 please enter a number between 1 and 10
1
For number 6 please enter a number between 1 and 10
7
For number 7 please enter a number between 1 and 10
1
For number 7 please enter a number between 1 and 10
9
```

Test 20

I tested the validation again but this time, it was to see what happens when player one inputs a letter after a number between 1 and 10. I made player input 3ee to see what happens. The programme should record this input as just 3 and continue.

```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy
For number 1 please enter a number between 1 and 10
3ee_
```

It worked as I expected it to.

```
C:\Users\shamu\source\repos\Project5\Debug\Project5.exe
Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower.
The number player one picked was 3. Is the next number higher or lower?
```

Problems

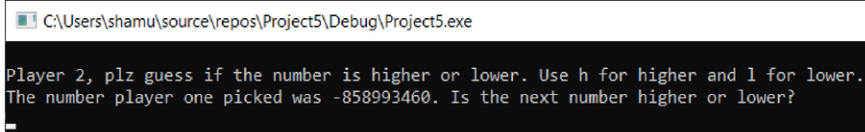
I have encountered various problems while doing task 1. Some of them was easier to solve than the other.

1. Displaying for player 1's 1st number for player 2

This was probably the easiest problem for me to solve. After I did the code for the while loop where I validated the input for player 1, the line saying what player 1s number wasn't working. It "-858993460" instead. I realised that this was because of the for-loop code. Instead of saying 0, and 6. I switched it to 1 and 7.

Before

```
for (count = 0; count < 6; count++) {  
    higherlower = guess(numbers, count);  
    if (higherlower == "h")
```

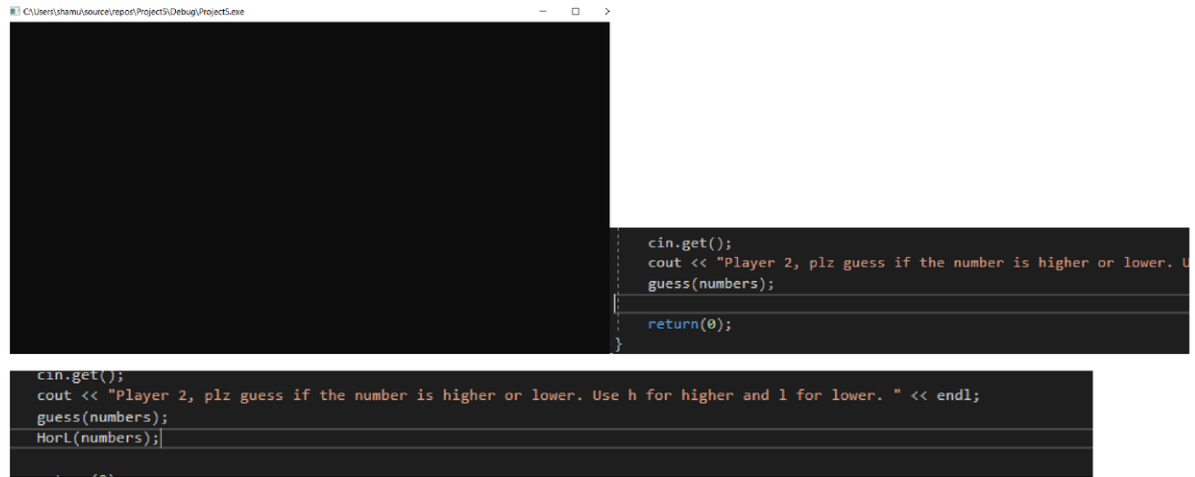


After

```
for (count = 1; count < 7; count++) {  
    higherlower = guess(numbers, count);  
    if (higherlower == "h")
```

2. Player 2 code doesn't run.

One problem that I have encountered while writing my code was that the code ended after it cleared the screen for player one's inputs. This meant that it did not show the instructions for player two and player two will not be able to guess if player one's numbers are higher or lower. It is just a black screen. I looked over my code again and realised that this was because I did not put add it to the main function. After finding that out, I added it on. This is probably the easiest problem I had to fix.



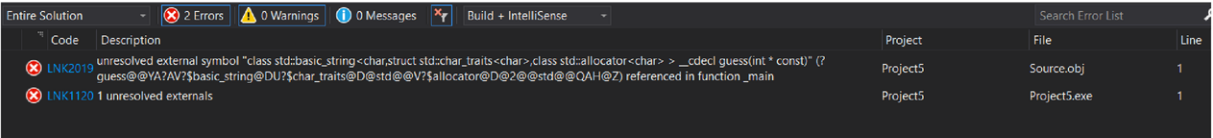
```
cin.get();
cout << "Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower. " << endl;
guess(numbers);
Horl(numbers);
// Horl(numbers);

return(0);
```

3. Unresolved externals

I faced another problem with my code for my game. Apprarntly I had an unresolved external symbol. At first I did not know what to do and it took me a while to figure out how to fix this To fix this.I clicked on the *LNK2019* and It sent me to a page on microsofts website. It says that I have this issue because I did not define a symbol. Apparantly. It can be caused by referencing a function. I added some lines of code for the guess function. This seemed to have done the trick. This was one of the harder problems as it took me a while to fix this issue. *LNK1120* was also fixed when I did this.

```
string guess(int numbers[])
{
    return string();
}
```



4. Validating player one's input

The hardest problem I faces was validating the user input. I had to make sure that the numbers player 1 input was numbers and they was between 1 and 2. My original code for it wasn't working so I had to change something.

The 1st code I had when I attempted the loop was this.

```
while (numbers < 1 || numbers > 10)
{
    cout << " please enter a number between 1 and 10" << endl;
    getline(cin, input);
    stream.str(input);
    stream >> numbers[count];
    stream.clear();
}
```

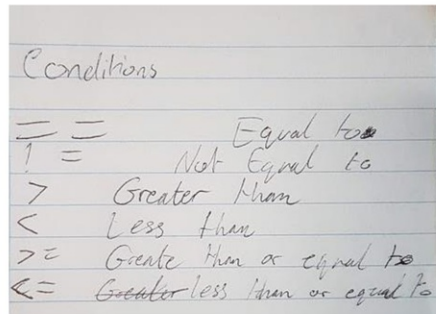
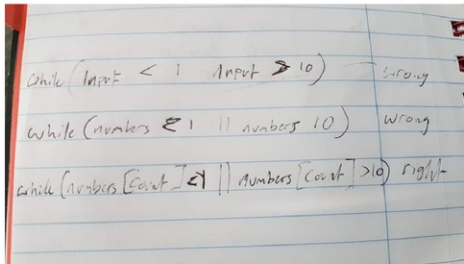
After doing some research I realised that I needed to add square brackets and add *count* in them. This should mean that player one will not get to the next number unless they enter a number between 1 and 10.

```
while (numbers[count] < 1 || numbers[count] > 10)
{
    cout << "For number " << count << " please enter a number between 1 and 10" << endl;
    getline(cin, input);
    stream.str(input);
    stream >> numbers[count];
    stream.clear();
}
```

This is shows that my code worked.

```
C:\Users\sham\source\repos\Project5\Debug\Project5.exe
Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower.
Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy
For number 1 please enter a number between 1 and 10
12
For number 1 please enter a number between 1 and 10
4
For number 1 please enter a number between 1 and 10
4
For number 2 please enter a number between 1 and 10
```

This is me working out the while line. It took me 3 tries to get it right(left) and me looking working out the conditions I will need to use(right).



Conclusion

My game is working well, I believe I have planned it all well and I also think I ran a good number of tests on the game. This experience has made improve my skills in coding with c++.

Appendix

Exercise 1 code

```
#include <string>
#include <sstream>
#include <iostream>

using namespace std;

void HorL(int numbers[]);

string guess(int numbers[]);

int main(int argc, char *argv[])
{
    string input;
    stringstream stream;

    int count;

    cout << "Player 1 should enter 7 numbers between 1 and 10 and then player 2 will guess if they are higher or lower." << endl;
    cout << "Player 2 wins by guessing all the numbers and player 1 wins after player 2 makes an incorrect guess. Enjoy" << endl;
    int numbers[7];
    for (count = 0; count < 7; count++) {
        cout << "please enter a number between 1 and 10" << endl;
        getline(cin, input);
        stream.str(input);
        stream >> numbers[count];
        stream.clear();
    }
    system("cls"); // clears the screen

    cin.get();
    cout << "Player 2, plz guess if the number is higher or lower. Use h for higher and l for lower. " << endl;
    guess(numbers);
    HorL(numbers);
}
```

```
        return(0);
    }

    string guess(int numbers[],int number)
    {
        string input;
        cout << "The number player one picked was " << numbers[number] << " Is the next number higher or lower?" << endl;
        getline(cin, input);
        return input;
    }

    void HorL(int numbers[]) {
        int count;
        string higherlower;

        for (count = 1; count < 7; count++) {
            higherlower = guess(numbers, count);
            if (higherlower == "h")
            {
                if (numbers[count] < numbers[count + 1])
                {
                    cout << "that was the right answer " << endl;

                }
                else
                {
                    cout << "that was the wrong answer" << endl;
                    cin.get();
                    break;
                }
            }
            else
            {
                if (numbers[count] >= numbers[count + 1])
                {
                    cout << "That was the right answer " << endl;
                }
                else
                {
                    cout << "that was the wrong answer " << endl;
                }
            }
        }
    }
}
```


FINAL GRADE

41 /100

GENERAL COMMENTS

Instructor

Exercise 1 (50) 41

Exercise 2 (30) 0

Exercise 3 (20) 0

There are a lot of tests provided for Exercise 1 but I wanted more evidence of the problem solving approach that you took.

The code contained a number of minor errors or coding issues.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7



Comment 1

What if they do not enter a value between 1 and 10?

You should test for this.

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29



Comment 2

Why not put in a separate function, this is a solution to a problem?

PAGE 30



Comment 3

Careful, you need to start with index = 0 to read the first array element

PAGE 31



Comment 4

How does player 2 win?



Comment 5

This does nothing, so it is pointless