

Software Development

by Jacob Goodrick

Submission date: 03-Apr-2019 09:18PM (UTC+0100)

Submission ID: 104002218

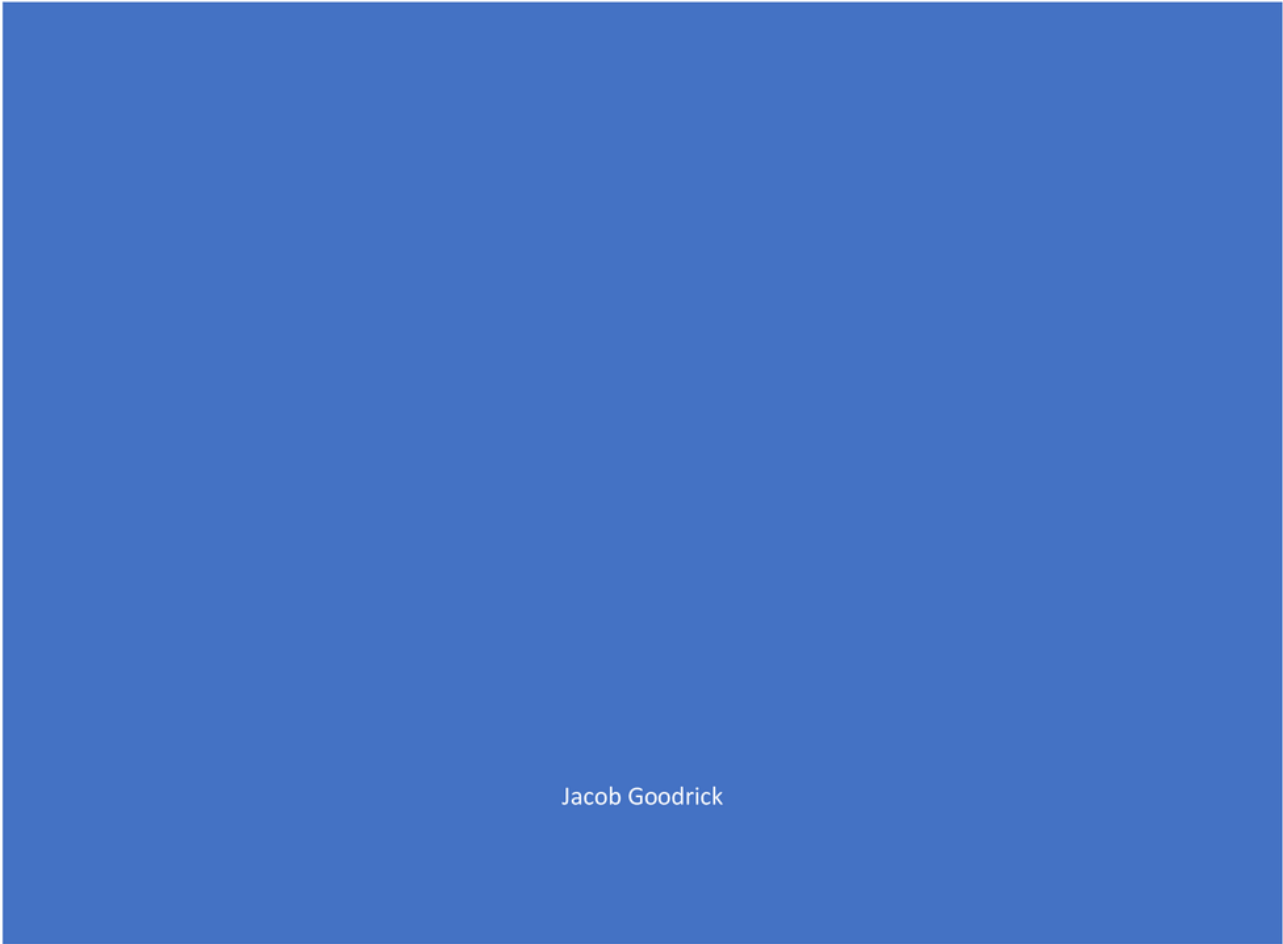
File name: 66561161.Software_Development_Assignment.docx (533.29K)

Word count: 3728

Character count: 17053



SOFTWARE DEVELOPMENT- PROGRAMING IN C



Jacob Goodrick

Contents

Higher or Lower Game	1
Introduction	1
Stages of Development	1
Problems Solved.....	4
Testing.....	4
Evaluation of the Higher or Lower game	6
Treasure Game.....	6
Introduction	6
Stages of Development	6
Problems Solved.....	10
Testing.....	11
Evaluation of the Treasure Game	13
Appendix	13
Higher or Lower Game Code.....	13
Treasure Game.....	15

Higher or Lower Game

Introduction

The first game that I am going to be making will be a very simple higher or lower game. The premise of the game is a 2-player game with the aim of one player trying to guess whether the numbers from the first player are either higher or lower. The game will consist of having to guess higher or lower on seven numbers, chosen by player 1. Then player 2 will have to get all seven numbers correct in order to win the game, if they get a single guess incorrectly then player 1 is the winner.

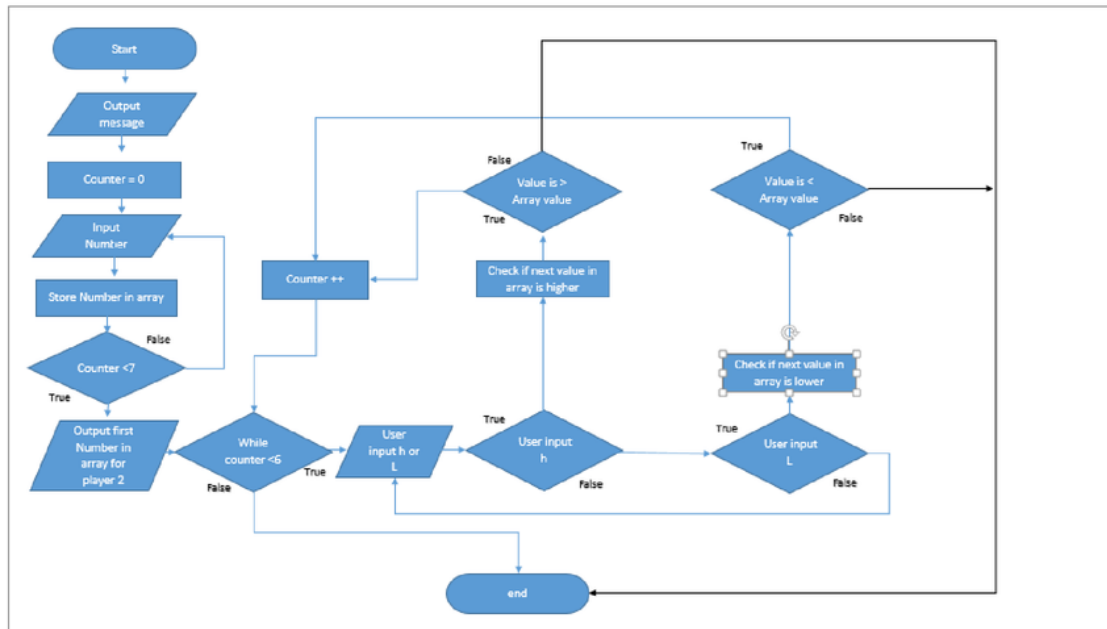
Stages of Development

Initial game steps:

1. Greet Players
2. Ask player 1 to input 7 numbers between 1-10
3. Display first value for player 2
4. Ask Player 2 to guess Higher or Lower

5. If player 2 is correct display next value
6. If player 2 is incorrect player 1 is the winner
7. If player 2 gets all values correct they are the winner

Basic Game Flowchart



This is an idea of what the game should look like in terms of a flowchart. The basic idea of the flowchart is that the player is going to input numbers into the player two will have to guess whether the next one is higher or lower. Which will be looped until either player 2's guess is incorrect or the counter hits six in which the game will end.

Development of the code

```

int Number[7]; //creating array with 7 spaces
int index; // counter
char guess; // variable for player 2's guesses
bool loss = true;
  
```

This is the start of my game. I started by thinking about the variables I would need for the game. I think it was important to get them out of the way first and if any more are required, I could add them later on.

Here I have created the array for the game, which contains seven spaces for each of the numbers player 1 will be inputting. The "int index" is being used for the for loop and if I need any other loops then I can be used there as well. Finally, "char guess" is a variable that I created for when player 2 is doing their guesses. I also have a Boolean true or false variable that is used for the loss conditions in the game.

```

cout << "Welcome to higher or lower game" << endl; //prints a welcome message to the player
cout << "Player 1 Please enter 7 numbers between 1-10" << endl; //asks the player to enter 7 numbers
    for(index = 0; index < 7; index++) //loop for inputting numbers into array. stops when the counter hits 7
    {
        cin>> Number [index] ; //inputting number into array
    }

cout << "Player 2 you are going to be guessing the numbers" << endl; //print player 2 message
cout << "The first number is" << endl; //prints message
index = 0; // setting counter to 0
cout << Number[index] << endl; // print the first number in the array

```

I started the coding of the game by creating outputs to display messages to the players. I used the outputs, as they are required to guide the players through the game telling them what they need to do. Here I created a "for loop" which I am using to put the numbers into the array.

After the loop I have created more messages for the players guiding them through the game while setting the index variable back to zero, and then finally printing the first variable entered by player 1 onto the screen so player 2 can see it.

```

while (index < 6) // while loop till the counter reaches greater than 6
{
    cout << "Is the next number higher or lower? "; //prints message
    cin >> guess; //player 2 input either h or l

    if (guess == 'h') // if statement for player 2 entering h
    {
        cout << "you guessed higher "; // prints message, what they have entered
        if(Number[index + 1] >= Number[index]) // if statement checking if the number is higher than number in array
        {
            cout << "You are Correct. " << endl; //prints player is correct
            cout << "The next value is " << Number[index + 1] << endl; //prints the next value in the array
            loss = false;
        }
        else //else for if the player gets the guess wrong
        {
            loss = true;
            if(loss == true) //if the player gets guess wrong this if will run
            {
                cout << "you have lost, player 1 the winner" << endl;
                break; //ends
            }
        }
        index++; //counter goes up by 1
    }
}

```

I have now started coding the player 2 section of the game which is them trying to guess whether player 1's number is higher or lower. Firstly I created a while loop which will only leave the loop once the index variable reaches six. I have done this because each time the player gets a guess correct the index counter increases until all of the values in the array have been shown.

After that I have created an if statement inside the while loop which will run when player 2 enters an "h". This will check if the next number in the array is higher if it is it will output correct message and output the next value. If not that, I have a true or false check for a loss, which outputs the message you have lost, which ends the game.

```

else if (guess == 'l') //if statement if player 2 enters l
{
    if (Number[index + 1] <= Number[index]) //if statement checking if the number is lower than the number in the array
    {
        cout << "You are Correct. " << endl; //prints player is correct
        cout << "The next value is " << Number[index + 1] << endl; //prints the next value on screen
        loss = false;
    }
    else //else for if the player guesses incorrectly
    {
        loss = true;

        if(loss == true) //if the player gets guess wrong this if will run
        {
            cout << "you have lost, player 1 the winner" << endl; //output message
            break; // ends
        }
    }
    index++; //adds to the counter
}
if(index == 6) //if counter reaches 6 player 2 is the winner
{
    cout << "player 2 wins"; //prints player 2 is winner
}

```

This is the code if player 2 enters an "l" this is the same as the last screen shot of code. At the bottom is an if statement that checks to see if the counter has reached 6 if it has then it runs the code telling player 2 they have won.

Problems Solved

During the development of the game, I had some issues with the player 2 being able to enter something other than "h" or "l" and the counter would increase, which would allow player 2 to win the game without actually needing to do any guesses. I found out that I had this problem because I had the counter going up by one in the incorrect place. The fix for this was very simple, I just required me to change the location of the index++ code so that I would only run when the player gets a correct guess with "l" or "h". Now if the player does enter a character that is not "h" or "l" it will just loop asking them to input higher or lower.

Throughout the development, I did have a few problems with the game. One of the problems I did come up with was how the game ended. Initially for player 2 winning the game I had it if you were to get the guess wrong it would just output a message saying you lost. This did not stop the game from running so the player could guess 6 times. What I did to fix this was adding a Boolean true or false check to the code and an if statement that would run when check is true, which would end the game.

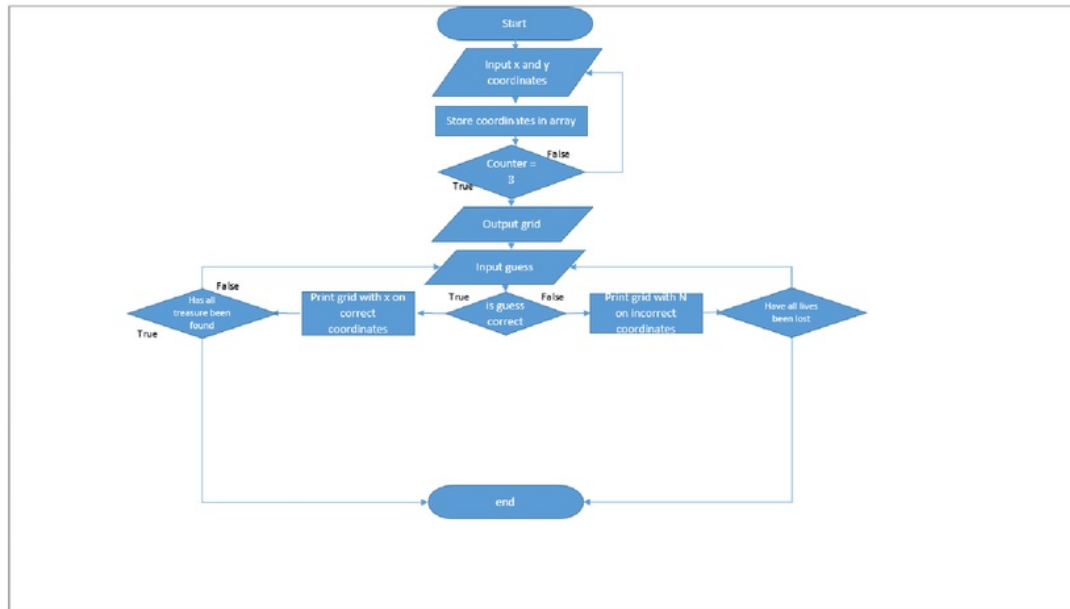
Testing

Test Number	what is being tested	Outcome	Evidence
-------------	----------------------	---------	----------

1	Inputting numbers 7,5,4,3,2,1,8	7,5,4,3,2,1,8 input into array, 7 should be the first number that is output as it is in the array.	Welcome to higher or lower game Player 1 Please enter 7 numbers between 1-10 7 5 4 3 2 1 8 Player 2 you are going to be guessing the numbers The first number is 7
2	Testing input for lower "l" being correct	Game should tell player they are correct and display next number "5"	Is the next number higher or lower? l You are Correct. The next value is 5 Is the next number higher or lower? l
3	Testing on 2 nd value	Game should tell player they are correct display 3 rd value	You are Correct. The next value is 4 Is the next number higher or lower? l
4	Testing on 3 rd value	Player is correct the 4 th value is displayed	You are Correct. The next value is 3 Is the next number higher or lower? l
5	Testing on 4 th value	Player is correct the 5 th value is displayed	You are Correct. The next value is 2 Is the next number higher or lower? l
6	Testing on 5 th value	Player is correct the 6 th value is displayed	You are Correct. The next value is 1 Is the next number higher or lower? h
7	Testing on 6 th value	player is correct, player 2 is the winner message display	Is the next number higher or lower? h you guessed higher You are Correct. The next value is 8 player 2 wins
8	Testing the player 2 winning condition	The game displays what the final value was and declares player 2 the winner	you guessed higher You are Correct. The next value is 8 player 2 wins
9	Testing the player 1 win condition	Player 2 gets the guess incorrect game displays player 1 is the winner	The next value is 4 Is the next number higher or lower? l you have lost, player 1 the winner

6. Print out grid with the guessed coordinates as a miss or a found
7. Find all 3 pieces of treasure
8. Alternatively, player 2 losses all lives.
9. Game ends depending on what step seven or eight occurs.

Treasure Game – Flowchart



Development of the code

```

int x; // declaring x coordinate
int y; // declaring y coordinate
int lives; // variable for lives
int index; // counter for game

char board[8][8]={          //array the game is going to played on
    {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0}};
  
```

Like the last game I wanted to try put in all the variables that I thought I would need to create the game. After spending time thinking about what I would need, I settled for these. I created an x and y variable

that I am going to use for storing the variable after being entered. I have a lives variable, which will be used for player 2's lives if they lose all three the game is over. Once again, I will be using an index variable, which is going to be my counter for game. I have also created char array, which is going to be where the player puts the treasure.

```
int main()
{

    cout << "Enter Treasure coordinates" << endl;    //output message
    cout <<"X - "; cin >> x;                          // input x coordinate
    cout <<"Y - "; cin >> y;                          // input y coordinate
    board[x][y]='T';                                // inputing x and y into array
    cout <<"Treasure has been placed" << endl;        //output message

    cout << "Enter Treasure 2 coordinates" << endl; //output message
    cout <<"X - "; cin >> x;                          // input x coordinate
    cout <<"Y - "; cin >> y;                          // input y coordinate
    board[x][y]='T';                                // inputing x and y into array
    cout <<"Treasure 2 has been placed" << endl;    //output message

    cout << "Enter Treasure 3 coordinates" << endl; //output message
    cout <<"X - "; cin >> x;                          // input x coordinate
    cout <<"Y - "; cin >> y;                          // input y coordinate
    board[x][y]='T';                                // inputing x and y into array
    cout << "Treasure 3 has been placed" << endl;    //output message
}
```

In the screenshot, I have just finished creating the input for my game. Here player 1 is going to input the x and y coordinates for three pieces of treasure. The pieces are treasure are classed as "T" so it will be easier for me to check for the array for the treasure later on in the game.

```
cout << "0 1 2 3 4 5 6 7 8" << endl; //printing the grid
cout << "1 - - - - - - - -" << endl;
cout << "2 - - - - - - - -" << endl;
cout << "3 - - - - - - - -" << endl;
cout << "4 - - - - - - - -" << endl;
cout << "5 - - - - - - - -" << endl;
cout << "6 - - - - - - - -" << endl;
cout << "7 - - - - - - - -" << endl;
cout << "8 - - - - - - - -" << endl;
```

Here I have created an output that is printing the grid onto the console; this is simply just to show the grid to the players.

```

while(lives < 3) //while loop until lives are 3
{
    cout << "Player 2 please enter an X and Y coordinate" << endl; //output message
    cin >> x; //input x coordinate
    cin >> y; //input y coordinate
    if('T' == board[x][y]) //if statement if guess is correct
    {
        cout << "well done you found Treasure" << endl; //output message

        board[x][y] = 2; //converting the correct coordinate into a 2 in the array

        cout << " 0 1 2 3 4 5 6 7" << endl; //print top row
        for(int y = 0; y < 8; y++) // counts y to less than 8
        {
            cout << y; //output y
            for(int x = 0; x < 8; x++) // checks all of the coordinates
            {
                if(board[x][y] == 2) // if statement for coordinate thats "2"
                {
                    cout << " X"; // outputs the coordinate as an x if its a "2"
                }
                else //outputss coordinates that are not a "2" as a "-."
                {
                    cout << " -.";
                }
            }
            cout << endl; //next line
        }
        index++; //counter adds 1
    }
}

```

This screenshot shows me completing the first section of the player 2 guessing coordinates. I have created a while loop that I have set to stay in the loop till the players lives reach 3 which should end the game as player 2 has lost.

This section includes the player 2 guessing, to start I have creating an input for the player to make a guess at the x and y coordinates where the treasure is. After that I have created an if statement that checks to see if the guesses are correct by the array for the letter "t" which is where the treasure has been placed. If the guess I have set it, up so that the game converts the "t" in the array to an integer "2". This is so I could use it in the for loop. The loop is used to print out the grid again but checks the array to see if there are any two values are stored there. If it does then the space in the grid is changed to and "X" when it prints out showing the player that they have found the treasure. For the other spaces in the array it will print it out as a "-." showing that nothing has been found at those coordinates.

```

else
{
    board[x][y] = 3; // converts incorrect coordinates to 3
    cout << "There is no Treasure here, you have lost a life" << endl; //output message

    cout << "  0  1  2  3  4  5  6  7" << endl; //prints top row
    for(int y = 0; y < 8;y++) //counts y to less than 8
    {
        cout << y; //output y
        for(int x = 0;x <8;x++) //counters x to less than 8
        {
            if(board[x][y] == 3) //if statement runs if coordinate in board is 3
            {
                cout << "  N"; //sets the incorrect incorrect guess as a N on the grid
            }
            else
            {
                cout << "  -"; //sets anything thats not the incorrect guess as "-"
            }
        }
        cout << endl; //next line
    }
    lives++; //lives counter adds 1
}
if(index == 3) //if statement if the index counter reaches 3
{
    cout << "Well done you found all of the Treasure." << endl; //output message
    break; //breaks from the if statement
}
if(lives == 3) //if statement that runs if the lives counter reaches 3
{
    cout << "You have lost all of your lives, you lose" << endl; //output message
}
}

```

The screenshot shows the end of the code. At the top of the code is an else, which will end up running if the player 2 guess is incorrect, the code that gets ran is similar to the code of the correct guess apart from a few different things. Firstly, the game will print out the grid and display the coordinates with an N to show the player where they incorrectly guess where the treasure is.

At the bottom, I have created two if statements, the first statement runs if the index counter reaches three, which means that player 2 has correctly guessed all of the treasure locations. Once this code runs the game will end, and tell player 2 they have won the game.

The other if statement is for the player 2's lives, due to the way I coded it the game keeps track of how many lives you have buy adding them from 0-3, which means at 0 the player has 3 lives, and at 3 the player has used all their lives. Once the player runs out of their lives the game will end, telling player 1 is the winner.

Problems Solved

Throughout the development of the game I did come up with a few problems, one of the easy fixes for the game was with inputting the treasure coordinates. Initially I wanted to have it loop to make it more

efficient, due to the way I implementing it, the method was not working. As a quick I decided to copy the code three times as it knew it would 100% work.

One of the other issues I had was trying to store the x and y coordinates. Originally, I wanted to save the treasure in the variable as "treasure". However, I was using the incorrect type of array. I had been using an integer array, which would not allow me to use a string, so instead of using the integer or string I decided on using a char, and only using single characters to store the values.

Testing

Test Number	what is being tested	Outcome	Evidence
1	Inputting coordinates for treasure 1	Saves coordinates x - 1 y - 1	<pre> Enter Treasure coordinates X - 1 Y - 1 Treasure has been placed </pre>
2	Inputting coordinates for treasure 2	Saves coordinates X - 3 Y - 3	<pre> Enter Treasure 2 coordinates X - 3 Y - 3 Treasure 2 has been placed </pre>
3	Inputting coordinates for treasure 3	Save coordinates X - 5 Y - 2	<pre> Enter Treasure 3 coordinates X - 5 Y - 2 Treasure 3 has been placed </pre>
4	Prints grid after inputting coordinates	Grid prints onto console	<pre> 0 1 2 3 4 5 6 7 8 1 - - - - - - - - 2 - - - - - - - - 3 - - - - - - - - 4 - - - - - - - - 5 - - - - - - - - 6 - - - - - - - - 7 - - - - - - - - 8 - - - - - - - - </pre>
5	Player 2 correct guess	Guesses x - 1 Y - 1	<pre> well done you found Treasure 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - X - - - - - 2 - - - - - - - 3 - - - - - - - 4 - - - - - - - 5 - - - - - - - 6 - - - - - - - 7 - - - - - - - </pre>
6	Location of treasure changes to "X" on grid.	X shows up on 1 - 1	<pre> well done you found Treasure 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - X - - - - - 2 - - - - - - - 3 - - - - - - - 4 - - - - - - - 5 - - - - - - - 6 - - - - - - - 7 - - - - - - - </pre>

7	Player 2 2 nd correct guess	Guess X - 3 Y - 3	<pre> well done you found Treasure 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - X - - - - - 2 - - - - - - - 3 - - - X - - - 4 - - - - - - - 5 - - - - - - - 6 - - - - - - - 7 - - - - - - - </pre>
8	Player 2 3 rd correct guess	Guess X - 5 Y - 2	<pre> well done you found Treasure 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - X - - - - - 2 - - - - X - - 3 - - - X - - - 4 - - - - - - - 5 - - - - - - - 6 - - - - - - - 7 - - - - - - - </pre>
9	Player 2 win message after finding all 3 pieces of treasure	Output message telling you found all of the treasure.	<pre> well done you found Treasure 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - X - - - - - 2 - - - - X - - 3 - - - X - - - 4 - - - - - - - 5 - - - - - - - 6 - - - - - - - 7 - - - - - - - Well done you found all of the Treasure. </pre>
10	Player 2 1 st incorrect guess	Guess X - 4 Y - 4	<pre> There is no Treasure here, you have lost a life 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - - - - - - - 2 - - - - - - - 3 - - - - - - - 4 - - - N - - - 5 - - - - - - - 6 - - - - - - - 7 - - - - - - - Player 2 please enter an X and Y coordinate </pre>
11	Player 2 2 nd incorrect guess	Guess X - 1 Y - 7	<pre> There is no Treasure here, you have lost a life 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - - - - - - - 2 - - - - - - - 3 - - - - - - - 4 - - - N - - - 5 - - - - - - - 6 - - - - - - - 7 - N - - - - - Player 2 please enter an X and Y coordinate </pre>
12	Player 2 3 rd incorrect guess	Guess X - 5 Y - 1	<pre> There is no Treasure here, you have lost a life 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - - - - N - - 2 - - - - - - - 3 - - - - - - - 4 - - - N - - - 5 - - - - - - - 6 - - - - - - - 7 - N - - - - - </pre>

13	Player 1 winner	Player 1 win message output	<pre> There is no Treasure here, you have lost a life 0 1 2 3 4 5 6 7 0 - - - - - - - 1 - - - - - N - 2 - - - - - - - 3 - - - - - - - 4 - - - - N - - 5 - - - - - - - 6 - - - - - - - 7 - N - - - - - You have lost all of your lives, you lose </pre>
----	-----------------	-----------------------------	---

Evaluation of the Treasure Game

After finishing the Treasure game, I am happy with the outcome of the game even though there are a few areas in which I could have improved. To start I think I could have tried to get the loop working for inputting the player 1 coordinates as it makes the code look quite messy and a bit less efficient. I managed to get everything working in the game apart from a few things such as trying to enter a letter into the coordinate, which breaks the game.

Appendix

Higher or Lower Game Code

```

#include <iostream>
#include <string>
#include <stdlib.h>
using namespace std;

int Number[7]; //creating array with 7 spaces
int index; // counter
char guess; // variable for player 2's guesses
bool loss = true;
int main()
{

cout << "Welcome to higher or lower game" << endl; //prints a welcome message
to the player
cout << "Player 1 Please enter 7 numbers between 1-10" << endl; //asks the
player to enter 7 numbers
    for(index = 0; index < 7; index++) //loop for inputting numbers into
array. stops when the counter hits 7
    {
        cin>> Number [index] ; //inputting number into array
    }

cout << "Player 2 you are going to be guessing the numbers" << endl; //print
player 2 message
cout << "The first number is" << endl; //prints message

```

```

index = 0; // setting counter to 0
cout << Number[index] << endl; // print the first number in the array

while (index < 6) // while loop till the counter reaches greater than 6
{
    cout << "Is the next number higher or lower? "; //prints message
    cin >> guess; //player 2 input either h or l

    if (guess == 'h') // if statement for player 2 entering h
    {
        cout << "you guessed higher "; // prints message, what they have entered
        if (Number[index + 1] >= Number[index]) // if statement checking if the
number is higher than number in array
        {
            cout << "You are Correct. " << endl; //prints player is
correct
            cout << "The next value is " << Number[index + 1] << endl;
//prints the next value in the array
            loss = false;
        }
        else //else for if the player gets the guess wrong
        {
            loss = true;
            if (loss == true) //if the player gets guess wrong this if will
run
            {
                cout << "you have lost, player 1 the winner" << endl;
                break; //ends
            }
        }
        index++; //counter goes up by 1
    }

    else if (guess == 'l') //if statement if player 2 enters l
    {
        if (Number[index + 1] <= Number[index]) //if statement checking if the
number is lower than the number in the array
        {
            cout << "You are Correct. " << endl; //prints player is correct
            cout << "The next value is " << Number[index + 1] << endl;
//prints the next value on screen
            loss = false;
        }
        else //else for if the player guesses incorrectly

```



```

cout << "Enter Treasure coordinates" << endl; //output message
cout <<"X - "; cin >> x; // input x coordinate
cout <<"Y - "; cin >> y; // input y coordinate
board[x][y]='T'; // inputing x and y into array
cout <<"Treasure has been placed" << endl; //output message

cout << "Enter Treasure 2 coordinates" << endl; //output message
cout <<"X - "; cin >> x; // input x coordinate
cout <<"Y - "; cin >> y; // input y coordinate
board[x][y]='T'; // inputing x and y into array
cout <<"Treasure 2 has been placed" << endl; //output message

cout << "Enter Treasure 3 coordinates" << endl; //output message
cout <<"X - "; cin >> x; // input x coordinate
cout <<"Y - "; cin >> y; // input y coordinate
board[x][y]='T'; // inputing x and y into array
cout << "Treasure 3 has been placed" << endl; //output message

cout << "0 1 2 3 4 5 6 7 8" << endl; //printing the grid
cout << "1 - - - - -" << endl;
cout << "2 - - - - -" << endl;
cout << "3 - - - - -" << endl;
cout << "4 - - - - -" << endl;
cout << "5 - - - - -" << endl;
cout << "6 - - - - -" << endl;
cout << "7 - - - - -" << endl;
cout << "8 - - - - -" << endl;

cout << "Player 2 you are going to be guessing where the Treasure is" << endl;
//output message

while(lives < 3) //while loop until lives are 3
{
    cout <<"Player 2 please enter an X and Y coordinate" << endl; //output
message
    cin >> x; //input x coordinate
    cin >> y; //input y coordinate
    if('T' == board[x][y]) //if statement if guess is correct
    {

```

```

    cout << "well done you found Treasure" << endl; //output message

    board[x][y] = 2; //converting the correct
coordinate into a 2 in the array

    cout << "  0  1  2  3  4  5  6  7" << endl; //print top row
    for(int y = 0; y < 8;y++) // counts y to less than 8
    {
        cout << y; //output y
        for(int x = 0;x <8;x++) // checks all of the
coordinates
            if(board[x][y] == 2) // if statement for
coordinate thats "2"
            {
                cout << "  X"; // outputs the coordinate
as an x if its a "2"
            }
            else
            { //outputss coordinates
that are not a "2" as a "-"
                cout << "  -";
            }

        cout << endl; //next line
    }

    index++; //counter adds 1
}
else
{
    board[x][y] = 3; // converts incorrect
coordinates to 3
    cout << "There is no Treasure here, you have lost a life" << endl;
//output message

    cout << "  0  1  2  3  4  5  6  7" << endl; //prints top row
    for(int y = 0; y < 8;y++) //counts y to less than 8
    {
        cout << y; //output y
        for(int x = 0;x <8;x++) //counters x to less than 8
            if(board[x][y] == 3) //if statement runs if
coordinate in board is 3
            {
                cout << "  N"; //sets the incorrect
incorrect guess as a N on the grid
            }
    }
}

```

```

    }
    else
    {
        cout << "  -";           //sets anything thats not
the incorrect guess as "-"
    }
    cout << endl;               //next line
}
lives++;                       //lives counter adds 1
}
if(index == 3)                 //if statement if the index counter
reaches 3
{
    cout << "Well done you found all of the Treasure." << endl;
//output message
    break;
//breaks from the if statement
}
if(lives == 3)
//if statement that runs if the lives counter reaches 3
{
    cout << "You have lost all of your lives, you lose" << endl;
//output message
}
}
}
}

```

Software Development

ORIGINALITY REPORT

5%

SIMILARITY INDEX

4%

INTERNET SOURCES

1%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Pembrokeshire College

Student Paper

1%

2

pubcrawler.org

Internet Source

1%

3

Submitted to Swansea Metropolitan University

Student Paper

1%

4

www.pucpr.br

Internet Source

1%

5

www.ar15.com

Internet Source

<1%

6

www.scribd.com

Internet Source

<1%

7

pastebin.com

Internet Source

<1%

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off

FINAL GRADE

GENERAL COMMENTS

50/100

Instructor

Exercise 1 29/ 50%

Flow chart shows some planning of task and conforms to conventions, however in future ensure all paths join boxes and there are no gaps. Your first loop from the numbers <7 false path needs an incremental counter process box and requires a reset back to 0 on the true path, as you indicate you will use the same variable to track where you are in the array holding the first players values whilst the second player attempts to predict if the next value is higher or lower. It would be improved if flowchart included output messages to players.

Code was well presented and annotated, use of functions would have improved readability, however your code was fairly efficient. A while statement with the range `number[index]<1 && number[index]>10` located immediately after your capture of this value would have been a way to prevent numbers outside of the range getting into the array. Whilst you acknowledge however this is not functioning, you should still have included tests for this feature as it was part of the assignment.

Exercise 2 21/ 30%

For the flowchart, again ensure all paths join boxes and there are no gaps. Your first loop collecting the coordinates needs an increment counter process box along the false path.

The flowchart does not take into account appropriate numerical range for inputs, to prevent stepping outside of treasure map, and does not indicate how a win or loss would occur, although your paths imply this, you should consider output process boxes on these paths indicating the win/loss.

A nested loop would be a faster way of initialising your two dimensional array.

When displaying your grid after a guess, you should have combined the code for both the correct and incorrect guesses using an if, else if, else structure, to cover your array containing both x's and N's .

Irrespective of if the guess was correct or not, once you updated the array with the appropriate x or N, then outside the if statement write the code to display the grid. What you have at the moment will only show the N's OR the x's, not both.

Again code does not validate for values outside of acceptable range.

Again as for previous exercise consider the use of functions.

Exercise 3 0/20%

You have not attempted this exercise.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19
