Project 1

<Tick-Tack-Toe>

CSC-5

Name: Alberto Garcia

Date: 7/14/2016

Introduction:

This is a tic-tac-toe game

The game lets the user either play against an AI or another Player

Whoever gets 3-in a row wins.

The user enters 1-9 to choose a spot:

For Example:

1	2	3		1	2	3
4	5	6		0	X	X
7	8	9		0	0	9

Summary:

Lines of code: 169

Comment lines (on side of code) : 79

Blank lines: 15

Amount of variables used:9

Number of Functions: 4

The project was quite tough. I spent about a week on it. The hardest part was drawing the map, for it was fairly tough referencing it to multiple voids. I decided to use a “Vector” because it was easier for me to reference it. This project included many of the concepts in the Savitch and Ghaddis Book’s

Description:

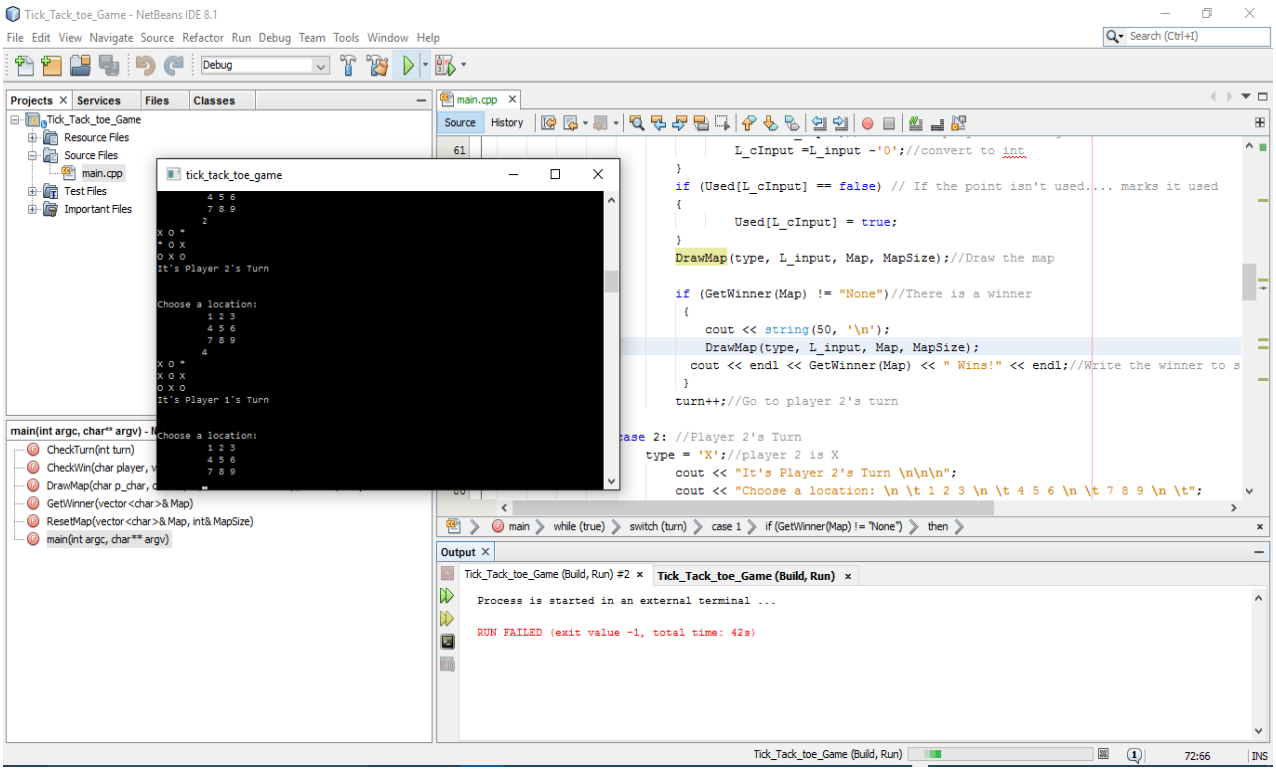
-The Game uses an array to store the input of the user.

For example: If the user picks “ 7 ”, Then (bool Used[7] = true;), and now the user cannot pick that spot anymore.

-The Game Logic works by looping through statements and constantly Checking if the user has 3 X’s or 3 O’s in a row.

It runs through an algorithm that checks all the possible spots to win.

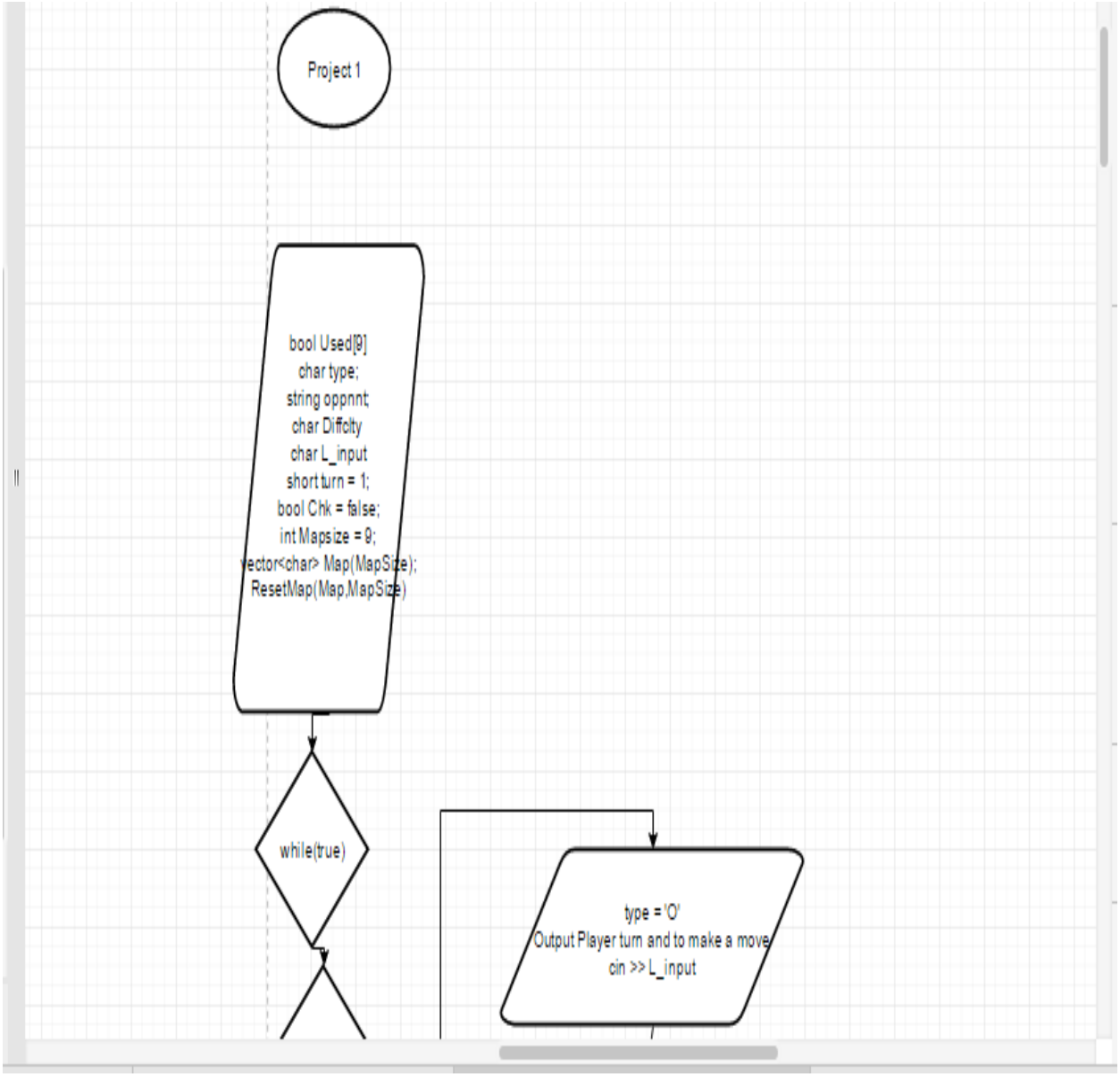
If the user enters a total of 5 spots, then it is a tie, because it is impossible to win after that since the whole map gets filled

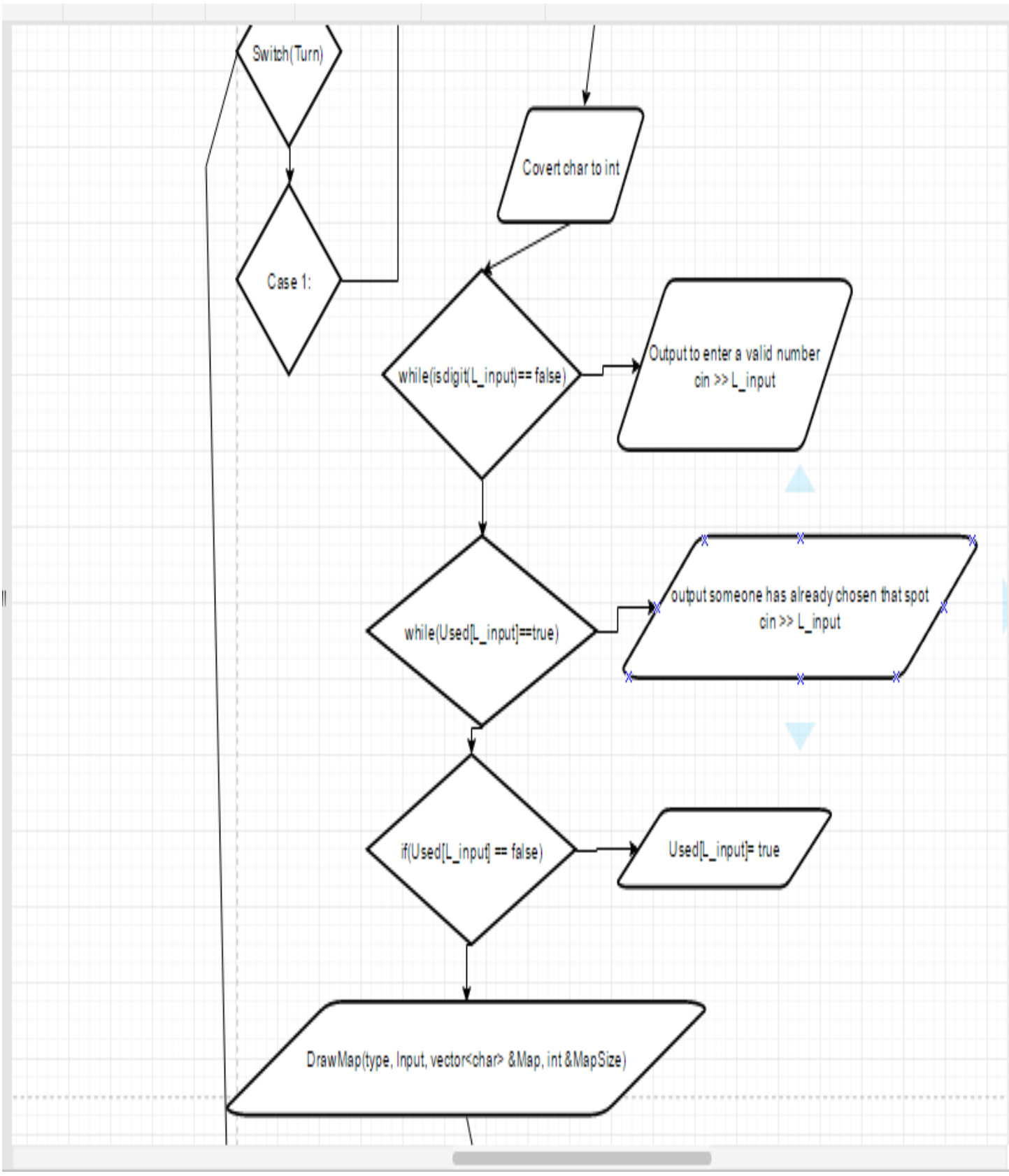


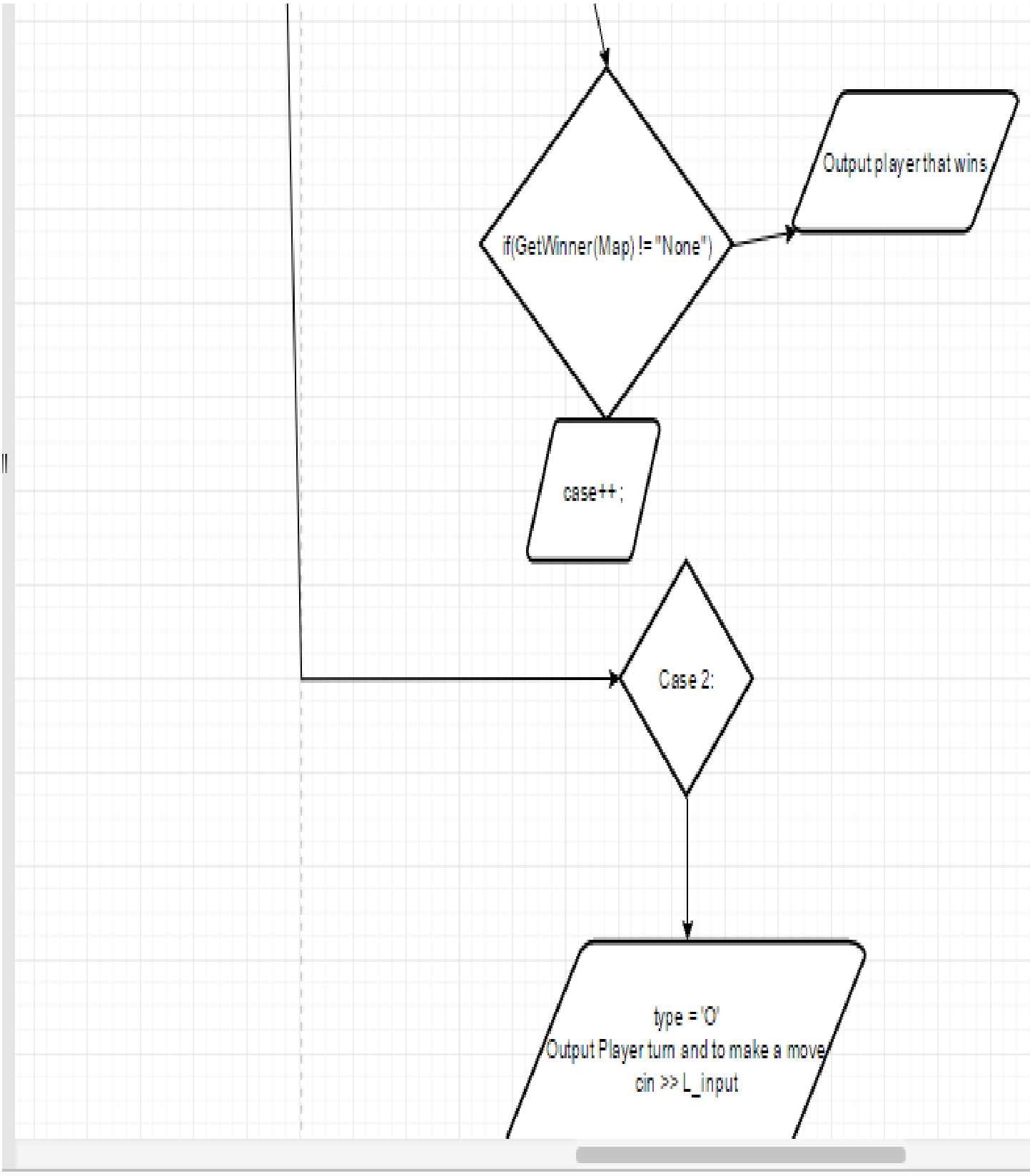
Function Prototypes:

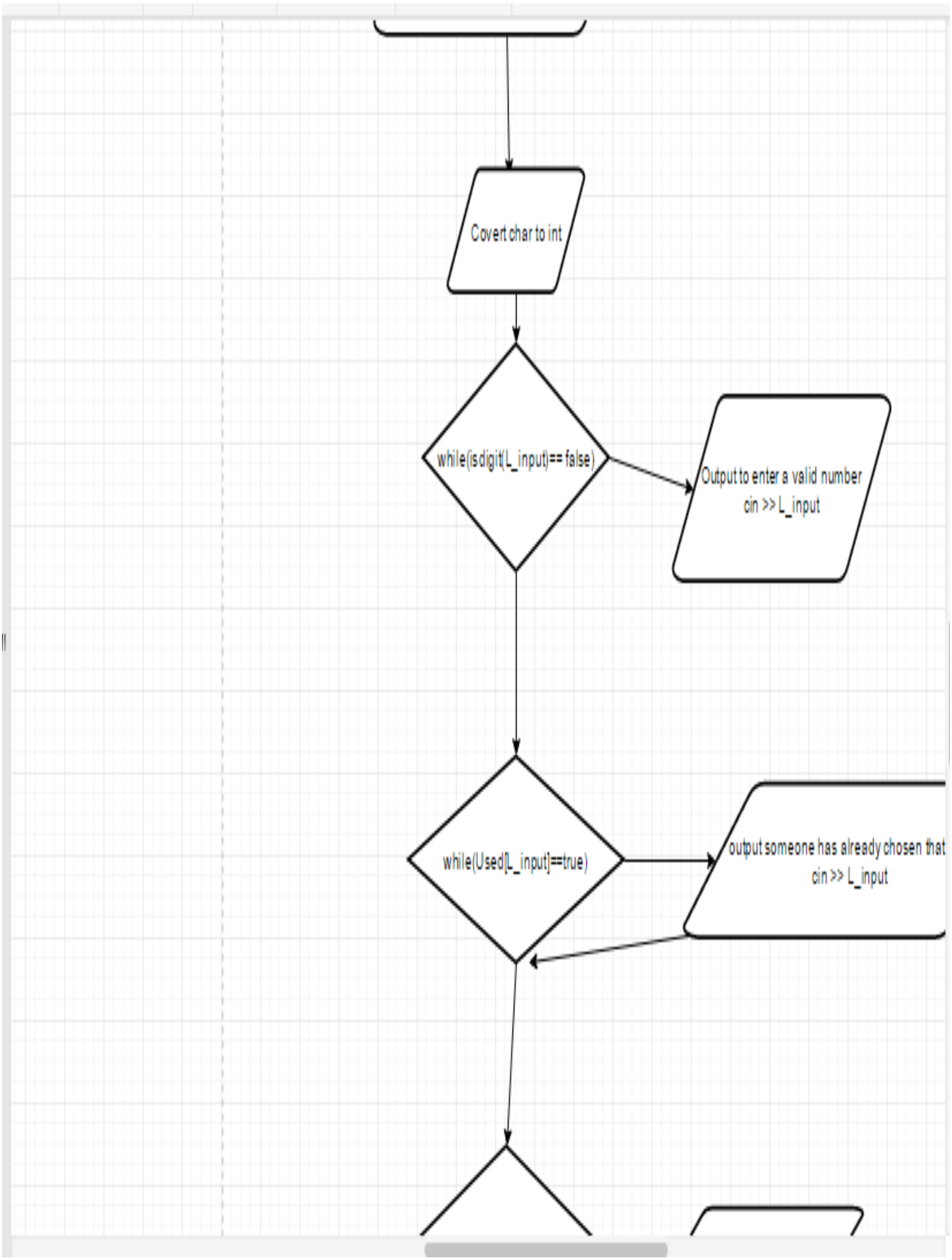
```
void DrawMap(char p_char, char Input, vector<char>) &Map, int &MapSize);
void ResetMap (vector<char> &Map, int &MapSize);
bool CheckWin(char player, vector<char> &Map);
string GetWinner(vector<char> &Map);
```

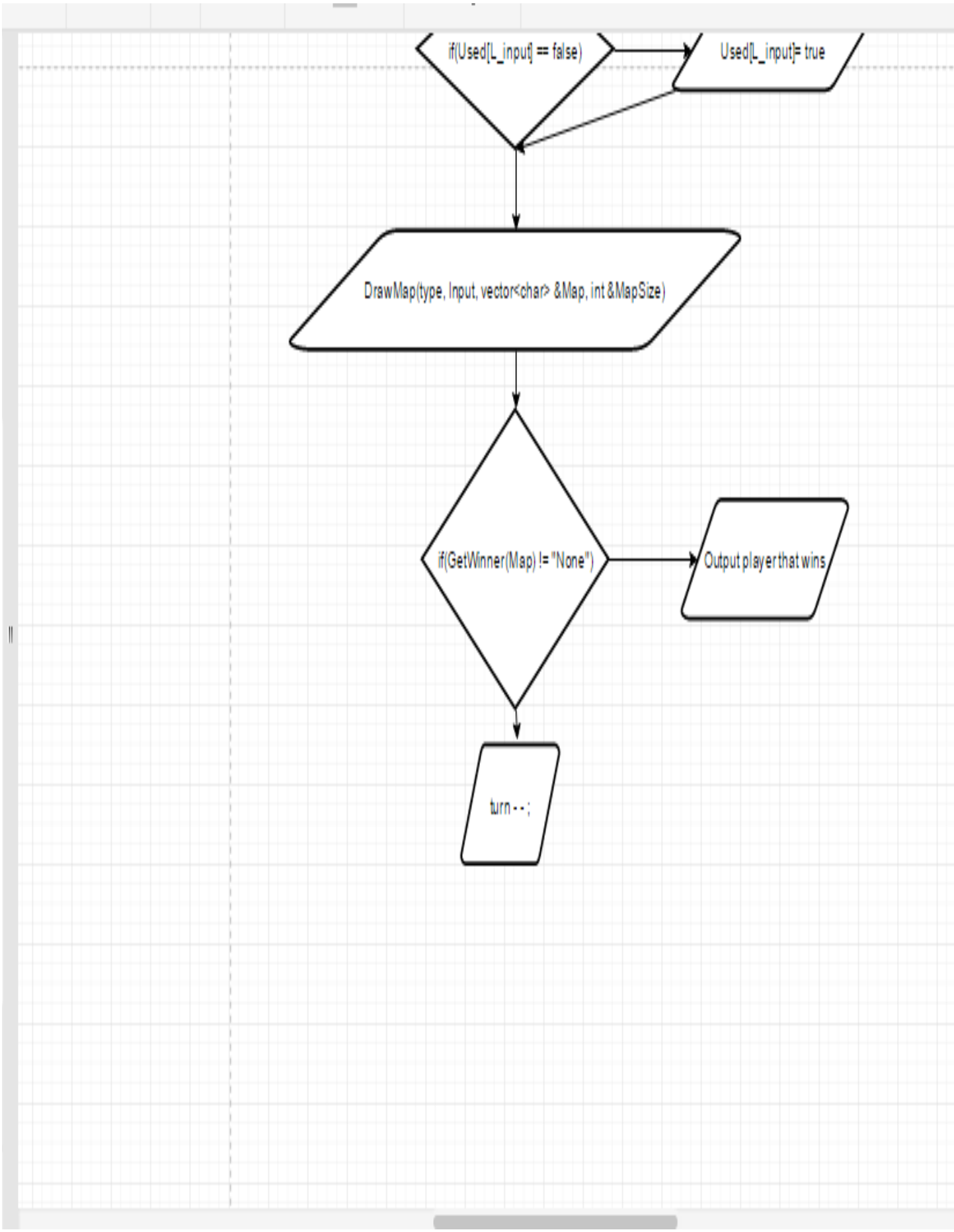
FlowChart: (Zoom in to see better!)




Switch(Turn)
Case 1:
Covert char to int
while(isdigit(L_input)== false)
Output to enter a valid number
cin >> L_input
while(Used[L_input]==true)
output someone has already chosen that spot
cin >> L_input
if(Used[L_input] == false)
Used[L_input]= true
DrawMap(type, Input, vector<char> &Map, int &MapSize)


if(GetWinner(Map) != "None")
Output player that wins
case++;
Case 2:
type = 'O'
Output Player turn and to make a move
cin >> L_input


Covert char to int
while(isdigit(L_input)== false)
Output to enter a valid number
cin >> L_input
while(Used[L_input]==true)
output someone has already chosen that
cin >> L_input


if(Used[L_input] == false)
Used[L_input]= true
DrawMap(type, Input, vector<char> &Map, int &MapSize)
if(GetWinner(Map) != "None")
Output player that wins
turn - - ;

Output of Code Logic Working:

Program Code:

```
/*
* File:   main.cpp
* Author: Alberto Garcia
* Created on July 14, 2016, 1:00 PM
* Purpose : Display A List Of Numbers TIC TAC TOE
*/

//System Libraries
#include <iostream> //Input/Output Library
#include <string> // Includes String Library
#include <vector> // Includes Vector
using namespace std; //Namespace of the System Libraries
 //User Libraries



 //Global Constants


 //Function Prototypes

void DrawMap(char p_char, char Input, vector<char> &Map, int &MapSize);
void ResetMap(vector<char> &Map, int &MapSize);
bool CheckWin(char player, vector<char> &Map);
string GetWinner(vector<char> &Map);

//Execution Begins Here!
int main(int argc, char** argv) {
bool Used[10]={};//What spot has been choose already
char type;//The current player character
char L_input;//the place the player chooses
short turn = 1;//whoose turn it is
int MapSize = 9;//Map y coordinate
       int L_cInput;//used for converting L_input to  int
vector<char> Map(MapSize);//Map for drawing tic tac toe
ResetMap(Map, MapSize);//Reset map

while (true)//loop forever
{
switch (turn)//check for turn
{
case 1: // Player 1's Turn
```

```
type = 'O';//player 1 is a O
cout << "It's Player 1's Turn \n\n\n";
cout << "Choose a location: \n \t 1 2 3 \n \t 4 5 6 \n \t 7 8 9 \n \t";
cin >> L_input;//where does the player want to go?
                    L_cInput = L_input - '0';//covert it to int
while (isdigit(L_input) == false)//is a digit?
{
cout << "Enter a valid number 1-9";
cin >> L_input;//where does player want to go?
                            L_cInput = L_input - '0';//convert it to int
}

while (Used[L_cInput] == true) // Checks if someone has already used the point
{
cout << "It appears someone has already chosen that spot..\t choose another spot\n";
cout << "\t 1 2 3 \n \t 4 5 6 \n \t 7 8 9 \n \t:";
cin >> L_input;//where does player want to go
                            L_cInput =L_input -'0';//convert to int
}
if (Used[L_cInput] == false) // If the point isn't used.... marks it used
{
Used[L_cInput] = true;
}
DrawMap(type, L_input, Map, MapSize);//Draw the map

                    if (GetWinner(Map) != "None")//There is a winner
        {
                        cout << string(50, '\n');
  cout << endl << GetWinner(Map) << " Wins!" << endl;//Write the winner to screen
        }
                    turn++;//Go to player 2's turn

              case 2: //Player 2's Turn
                 type = 'X';//player 2 is X
cout << "It's Player 2's Turn \n\n\n";
cout << "Choose a location: \n \t 1 2 3 \n \t 4 5 6 \n \t 7 8 9 \n \t";
cin >> L_input;//where does player want to go?
                    L_cInput = L_input - '0';//convert to int

while (isdigit(L_input) == false)//is it a digit?
{
cout << "Enter a valid number 1-9";
cin >> L_input;//where does player want to go?
                            L_cInput = L_input - '0';//covert to int
}
while (Used[L_cInput] == true) // Checks if someone has already used the point
```

```
{
cout << "It appears someone has already chosen that spot..\t choose another spot\t";
cout << "1 2 3 \n \t 4 5 6 \n \t 7 8 9 \n \t:";
cin >> L_input;//where does player want to go?
                        L_cInput = L_input - '0';//convert to int
}
if (Used[L_cInput] == false) // If the point isn't used.... marks it used
{
Used[L_cInput] = true;
}
DrawMap(type, L_input, Map, MapSize);//Draw the map

                if (GetWinner(Map) != "None")//There is a winner
        {
                    cout << string(50, '\n');
  cout << endl << GetWinner(Map) << " Wins!" << endl;//Write the winner to screen

        }
                turn--;//Go to player 1's turn
         }



}

//Exit Stage Right!
return 0;
}
void DrawMap(char p_char, char Input, vector<char> &Map, int &MapSize)//Requires a
return of player character and the position he wants to place it
{
int uInput = Input - '0'; uInput -= 1;//Convert char to int

for (int i = 0; i < MapSize; i++)
{
if (i % 3 == 0 && i != 0) { cout << endl; }//Add line skips when at 3

Map[uInput] = p_char;//assign the player character(x or o) to the map
cout << Map[i] << " ";//Draw the whole Map
}
cout << endl;//Skip a line

}
void CheckTurn(int turn)
{
cout << "Player " << turn << "'s turn" << endl; //write the current players turn
```

```
}

void ResetMap(vector<char> &Map, int &MapSize)
{
for (int i = 0; i < MapSize; i++)//Reset the map to empty spaces
{
Map[i] = '*';//Fill in map
}
}
bool CheckWin(char player, vector<char> &Map)//Checks for a winner
{
string Win = "";//The winner is stored here
vector<string> map(9);//a temperarely stored map used for conversion of char to string
for (int i = 0; i < 9; i++)//loop the map length
{
map[i].push_back(Map[i]);//set the string map equal to the char map
}
Win.push_back(player); Win.push_back(player); Win.push_back(player);//Add 3 chars
of itself so that it can compare to 3 places at once
return ((map[0] + map[1] + map[2] == Win) //Row 1
|| (map[3] + map[4] + map[5] == Win) //Row 2
|| (map[6] + map[7] + map[8] == Win) //Row 3
|| (map[0] + map[3] + map[6] == Win) //Column 1
|| (map[1] + map[4] + map[7] == Win) //Colomn 2
|| (map[2] + map[5] + map[8] == Win) //Column 3
|| (map[0] + map[4] + map[8] == Win) //Diagnol 1
|| (map[2] + map[4] + map[6] == Win)); //Diagnol 2
}
string GetWinner(vector<char> &Map)//Grab a winner and return string
{
if (CheckWin('O', Map)) return "Player 1";//if a winner is x return player 1
if (CheckWin('X', Map)) return "Player 2";//if winner is o return player 2
else return "None";//if there is no winner return non
}
```

Primitive Data Types

Type	Variable Name	Description	Location
Integer	MapSize	Is used as a reference to the Map Size	Main () Line: 34
Integer	L_cInput	Used for converting L input to int	Main() Line: 35
Integer	uInput	Stores User Input	Void DrawMap() Line: 36
Short	turn	Tracks which player's turn it is	Main() Line: 33
Char	type	Used to determine whether the player is 'x' or 'o'	Main() Line: 31
Char	L_input	Used to track the move the player made last	Main() Line: 32
Bool	Used[10]	Stores all the moves the player has made so they	Main() Line: 30

		won't make the same move	

Constructs

Chapter	Keywords	Location Samples

2
 	
Line: 56
	Equality Operators !=,==,>,<>=,=<, operators	While(Used[L_cInput]==true)
		{
		output
		input
		}

3
 	
Line: 124
	Nested loop	For(int i = 0; i <MapSize;i++)
		{
		If(i%3 == 0&&i!=0) { }
		}

5 **Line: 67**	Call by value Function parameter(char x)	DrawMap(type, L_input, Map, MapSize);
5 **Line: 37**	Call by reference Void(&x)	ReserMap(vector<char> &Map, int &MapSize);

File io's && formatting

Chapter	**Location Sample**	**Description**
6 **Line: 47**	Cin >> L_input	User input that tracks the spot the user chooses
6 **Line: 45**	Cout <<"It's Player 1's Turn \n\n\n"	Outputs who's Turn it is and skips 3 lines
6 **Line: 71**	Cout<< string (50,'\n')	Skips 50 lines after player wins
6 **Line: 46**	Cout << "Choose a location:\t 1 2 3 "	Outputs to put a location and then tabs