Tic Tac Toe Game

Tic Tac Toe Game

- Name: Soumil Nitin Shah
- ID#: 1031686
- Program 1
- Due Date: Sept 16, 2019,
- Date of Submission: Sept 14 2019
- Website: https://soumilshah.herokuapp.com

Name: Soumil Nitin Shah
ID#: 1031686
Program 1
Due Date: Sept 16, 2019
Date of Submission: 14 September 2019
Github: https://github.com/soumilshah1995/Tic-Tac-Toe-Game-in-C-plus-plus#step-2

Table of Contents

Challenges

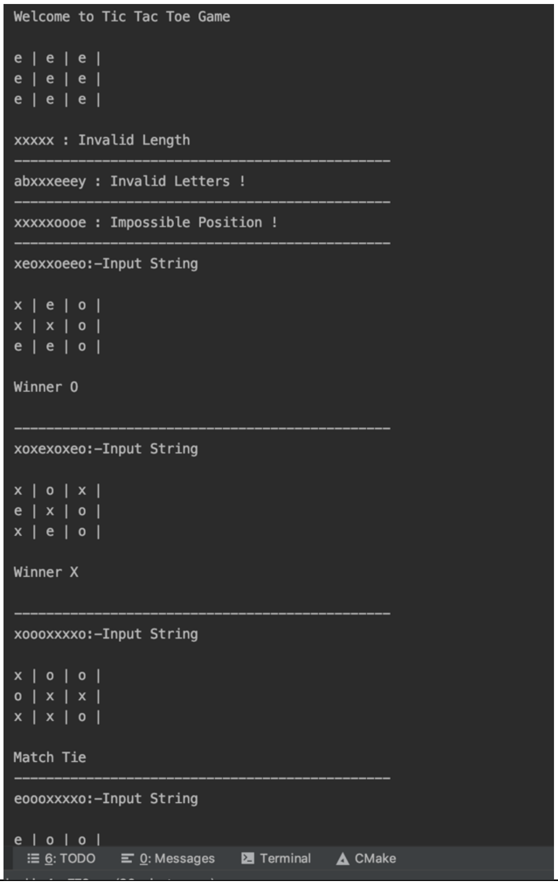
1. **Had lot of difficulty in making logic for Validation**
2. **Had issue how to convert Vector which has Empty space position how to choose random value from vector of empty spaces**
3. **Implementing Logic for if Game is Continue How to predict he Next Move**
4. **With Operator Overloading**

Main File

Main.cpp

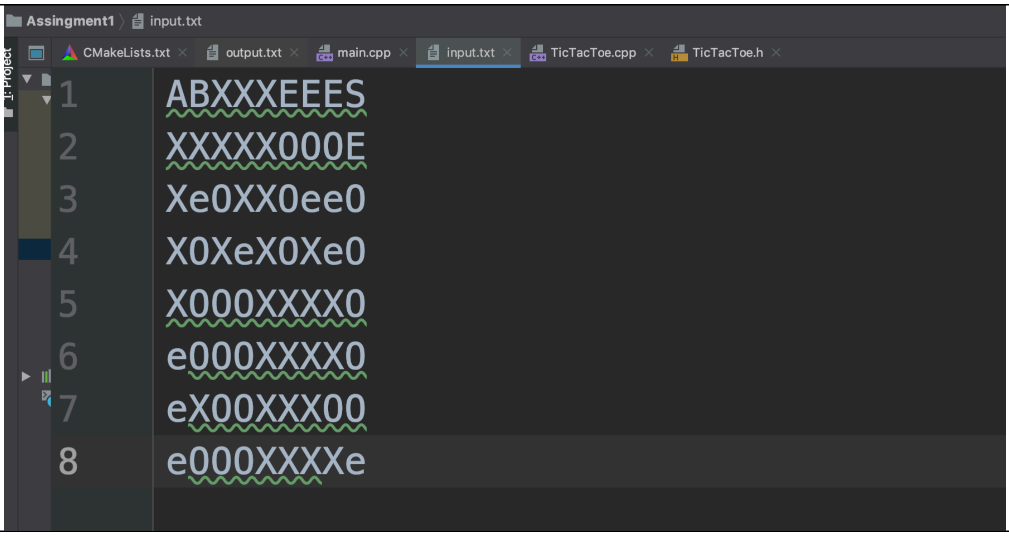
```
/*
 * Main .CPP File
 * Tic Tac Toe Game
 * Author : Soumil Nitin SHah
 * Version 0.0.1
 * Email soushah@my.bridgeport.edu
 * Student id : 1031686
 * Github:
 * Last Modified : 14 Sepetember 2019
 *
 * Operating System: MAC os
 * Configured with: --prefix=/Applications/Xcode.app/Contents/Developer/usr --with-
gxx-include-
dir=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/S
DKs/MacOSX10.14.sdk/usr/include/c++/4.2.
 * Apple LLVM version 10.0.1 (clang-1001.0.46.3)
 * Target: x86_64-apple-darwin18.7.0
 * InstalledDir:
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/b
in
 */

#include <iostream>
#include <fstream>
#include "TicTacToe.h"

using namespace std;

int main() {

    TicTacToe ttt;

    string pos;
    ifstream fin ("/Users/soumilshah/CLionProjects/TicTacToeGame/input.txt");
    ofstream fout("output.txt");

    while (!fin.eof()) {
        getline(fin, pos);
        ttt.setPosition(pos);
        cout << ttt;
        fout << ttt;

    }
    fout.close();

    return 0;
}
```

Screen Shots



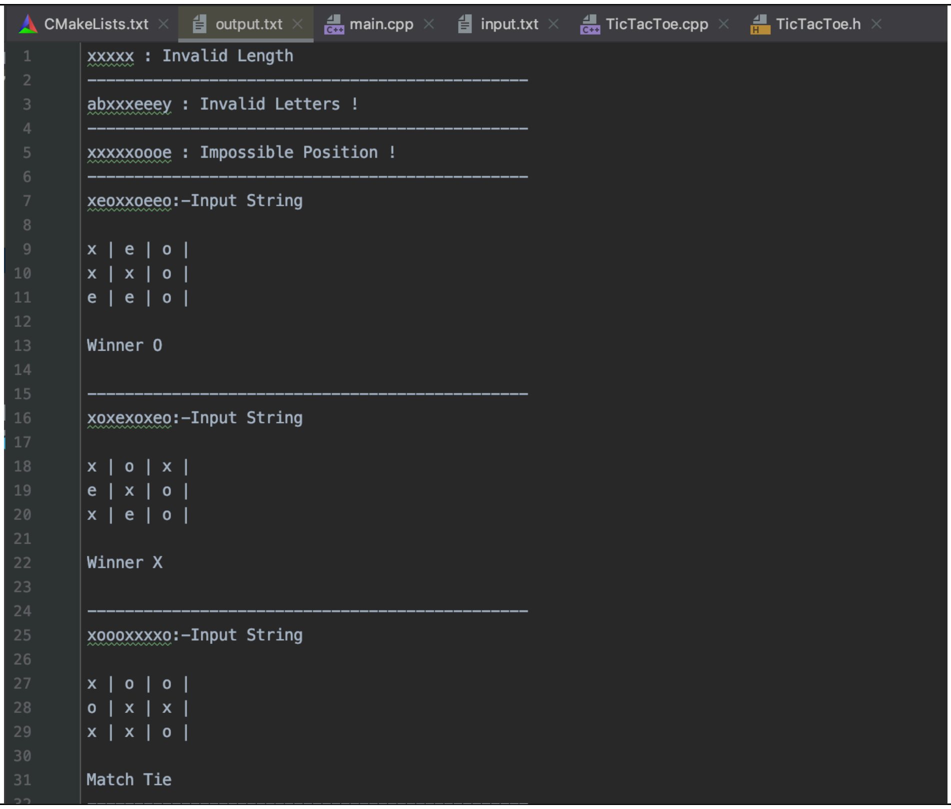
```
Welcome to Tic Tac Toe Game

e | e | e |
e | e | e |
e | e | e |

xxxxx : Invalid Length
-----------------------------------------------
abxxxeeey : Invalid Letters !
-----------------------------------------------
xxxxxoooe : Impossible Position !
-----------------------------------------------
xeoxxoeeo:-Input String

x | e | o |
x | x | o |
e | e | o |

Winner O

-----------------------------------------------
xoxexoxeo:-Input String

x | o | x |
e | x | o |
x | e | o |

Winner X

-----------------------------------------------
xoooxxxxo:-Input String

x | o | o |
o | x | x |
x | x | o |

Match Tie
-----------------------------------------------
eoooxxxxo:-Input String

e | o | o |
```

Figure shows Console Screen Shots



```
ABXXXEEES
XXXXX000E
Xe0XX0ee0
X0XeX0Xe0
X000XXXX0
e000XXXX0
eX00XXX00
e000XXXXe
```

Figure 2 shows inputs Txt file Screen shot opened on Clion Mac OS



```
xxxxx : Invalid Length
-----------------------------------------------
abxxxeeey : Invalid Letters !
-----------------------------------------------
xxxxxoooe : Impossible Position !
-----------------------------------------------
xeoxxoeeo:-Input String

x | e | o |
x | x | o |
e | e | o |

Winner O


-----------------------------------------------
xoxexoxeo:-Input String

x | o | x |
e | x | o |
x | e | o |

Winner X


-----------------------------------------------
xoooxxxxo:-Input String

x | o | o |
o | x | x |
x | x | o |

Match Tie
```

Figure 3 shows Output Txt file Screen shot opened on Clion Mac OS Text editor

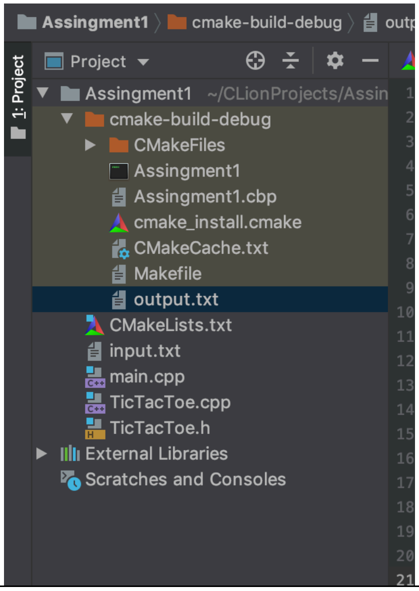


Figure 4 shows project Structure

References

1. Stack overflow
Link: https://stackoverflow.com/questions/21516575/fill-a-vector-with-random-numbers-c

This link was used as a reference how to get random Number from a vector I have taken the concept from this and implemented in my project

2. YouTube Videos on knowing how Tic Tac Toe Actually works what should be the Logic Behind

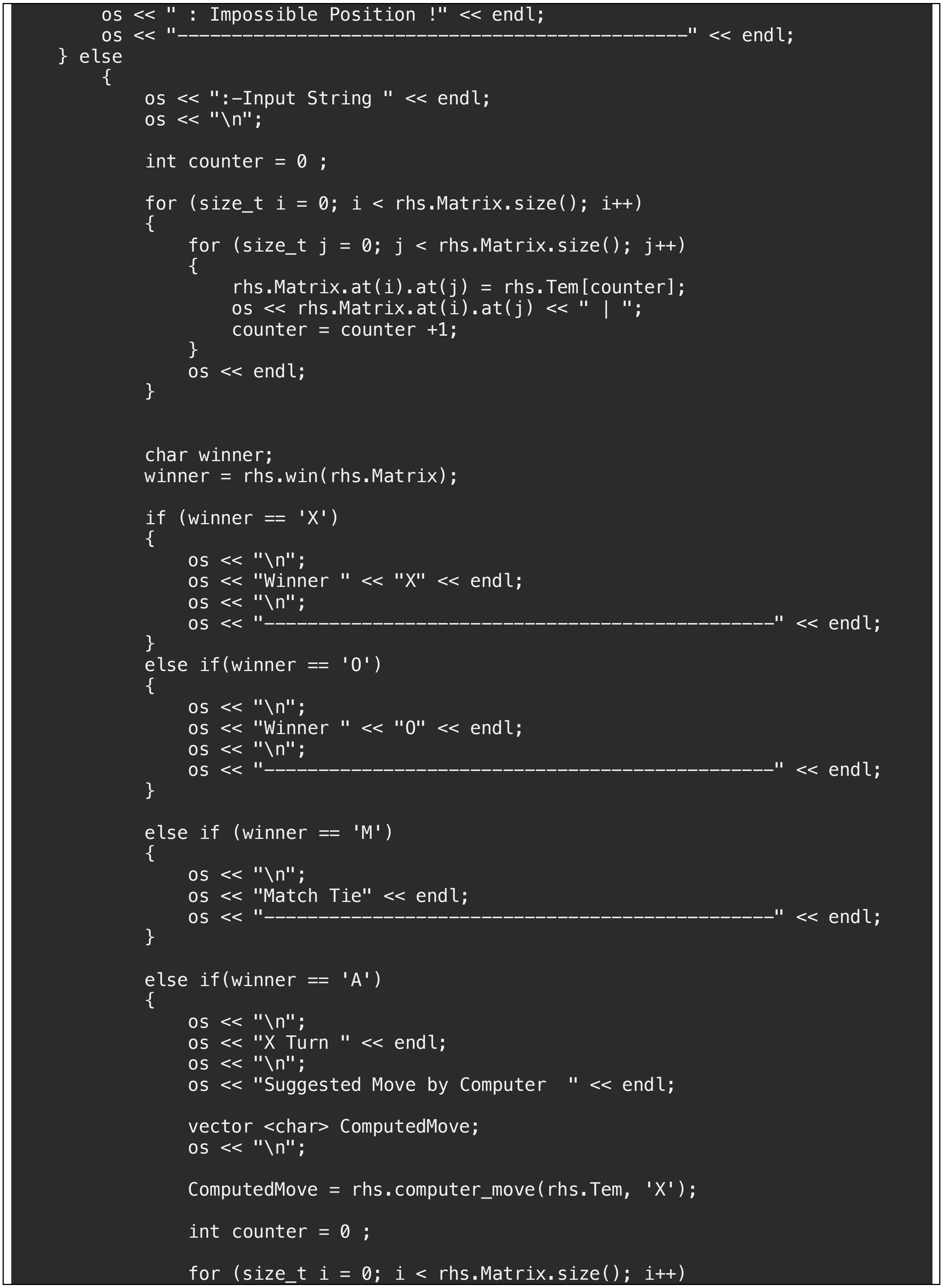
Link https://www.youtube.com/watch?v=xwwl8TgkwgU
Link : https://www.youtube.com/watch?v=F_npoo9AEKU&t=344s

3. Generating random Numbers
Link: https://stackoverflow.com/questions/13445688/how-to-generate-a-random-number-in-c

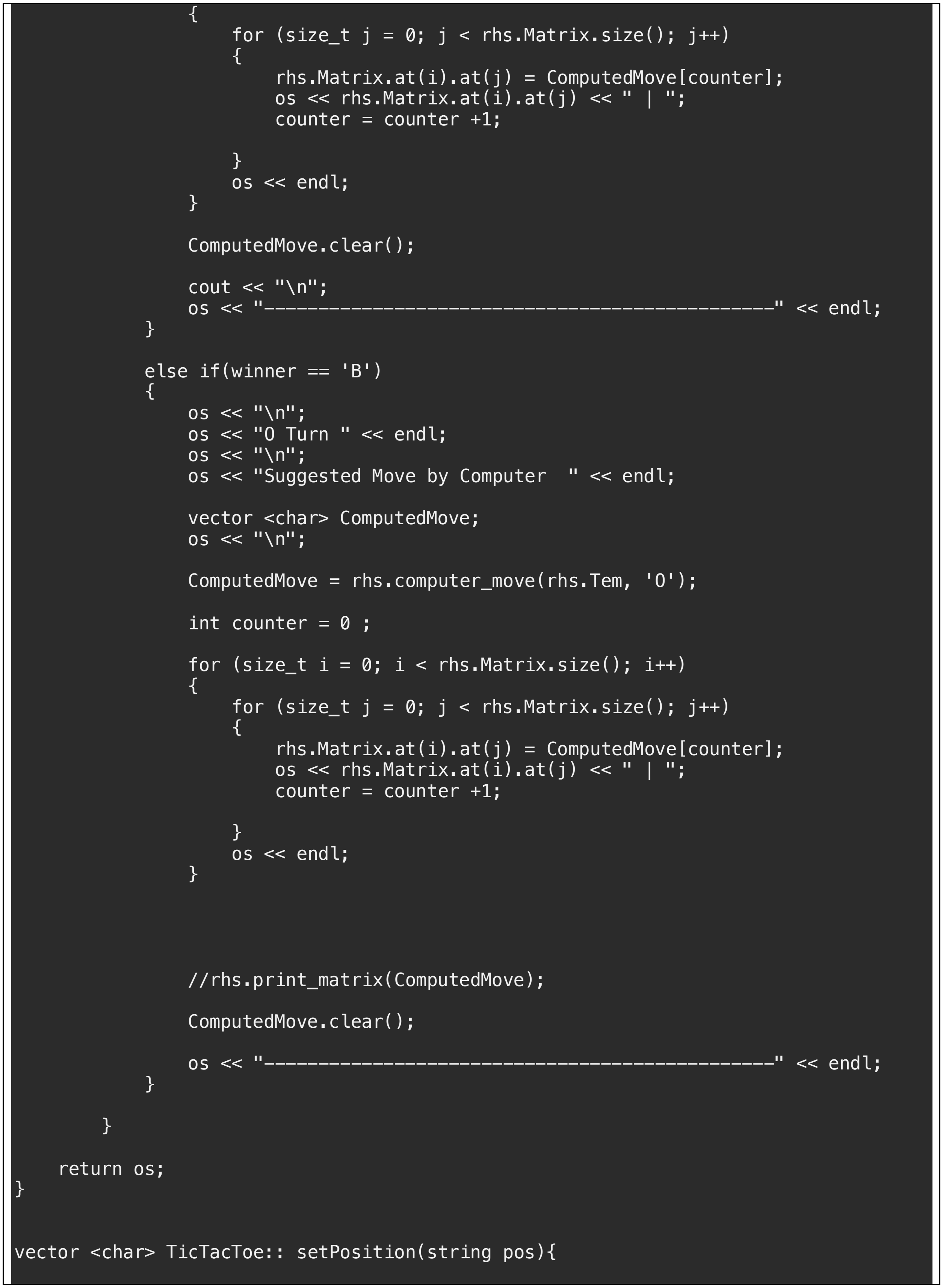
Source Code:

TicTacToe.cpp

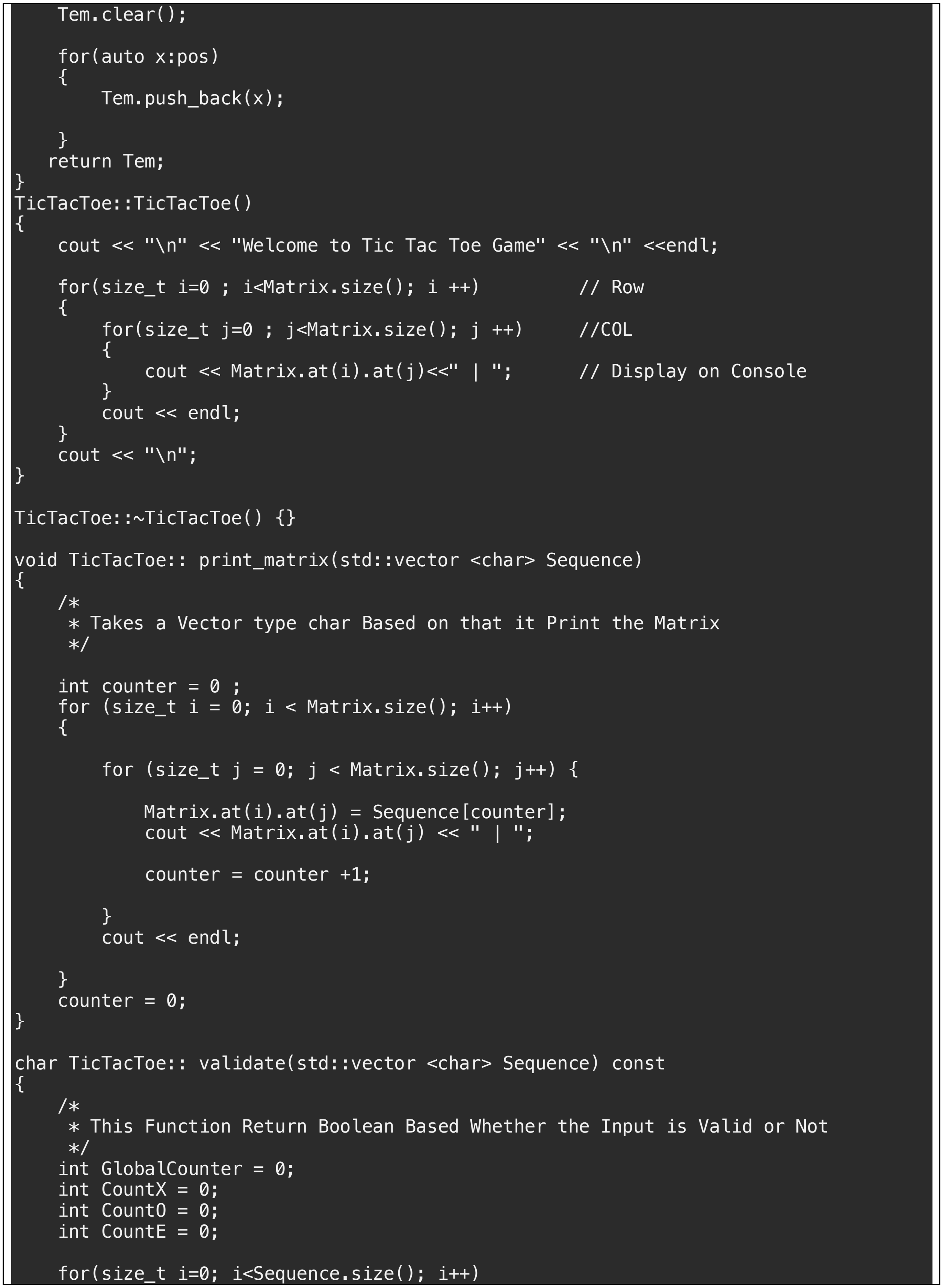
```
/*
 * TicTacToe.CPP File
 * Tic Tac Toe Game
 * Author : Soumil Nitin SHah
 * Version 0.0.1
 * Email soushah@my.bridgeport.edu
 * Student id : 1031686
 * Github:
 * Last Modified : 14 Sepetember 2019
 *
 * Operating System: MAC os
 * Configured with: --prefix=/Applications/Xcode.app/Contents/Developer/usr --with-gxx-include-dir=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.14.sdk/usr/include/c++/4.2.
 * Apple LLVM version 10.0.1 (clang-1001.0.46.3)
 * Target: x86_64-apple-darwin18.7.0
 * InstalledDir:
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin
 */

#include "TicTacToe.h"
#include <fstream>
#include <vector>
#include <string>
#include <iostream>
#include <ctime>
using namespace std;


std::ostream &operator << (std::ostream &os,  TicTacToe rhs)
{
    for(auto x :rhs.Tem)
    {
        os << x;
    }

    char value;
    value = rhs.validate(rhs.Tem);

    if (value == 'A')
    {
        os << " : Invalid Letters ! " << endl;
        os << "-----------------------------------------------" << endl;
    }

    else if(value == 'B')
    {
        os << " : Invalid Length " << endl;
        os << "-----------------------------------------------" << endl;
    }

    else if(value == 'C')
    {
```



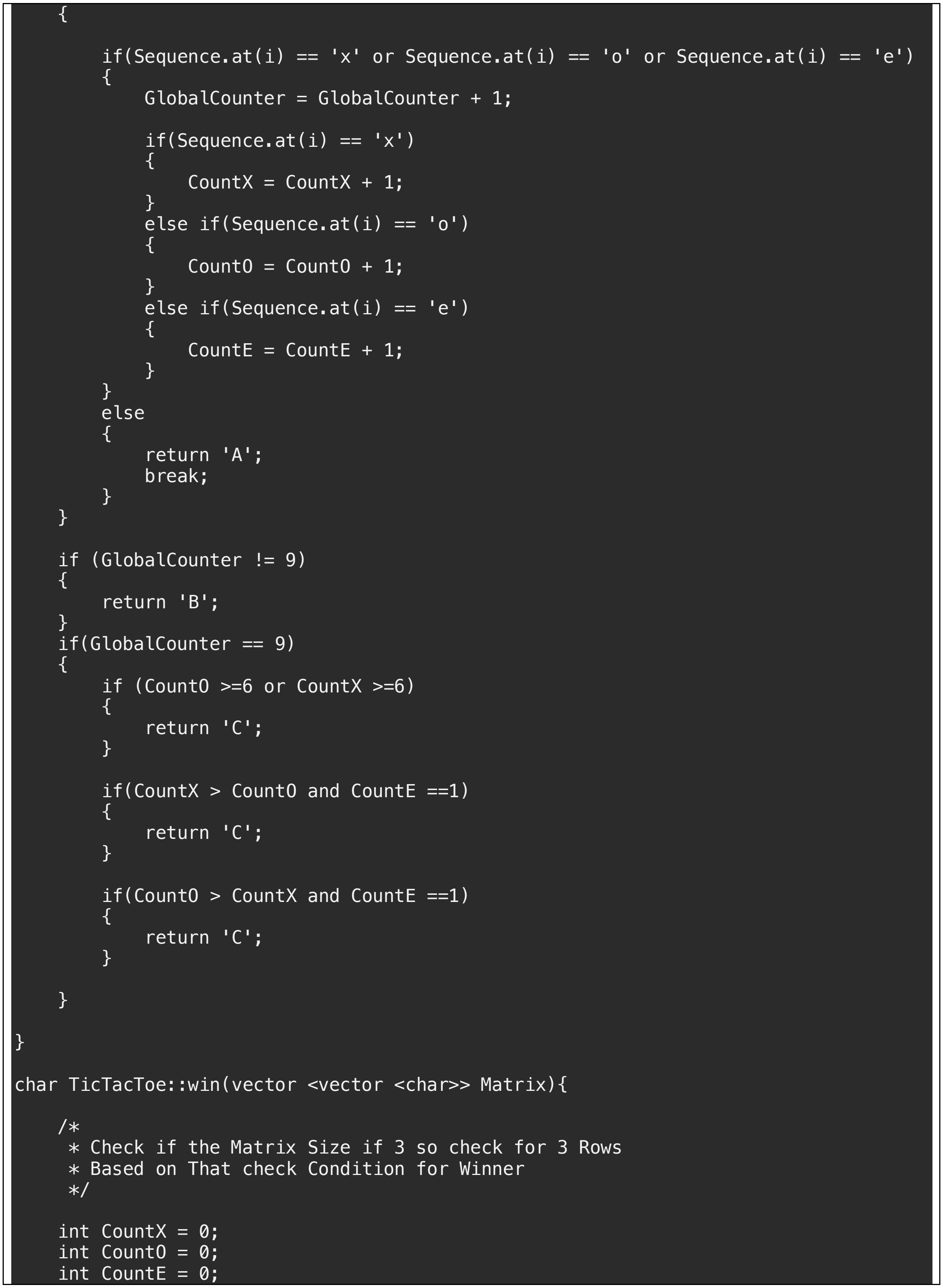
```
        os << " : Impossible Position !" << endl;
        os << "-----------------------------------------------" << endl;
    } else
        {
            os << ":-Input String " << endl;
            os << "\n";

            int counter = 0 ;

            for (size_t i = 0; i < rhs.Matrix.size(); i++)
            {
                for (size_t j = 0; j < rhs.Matrix.size(); j++)
                {
                    rhs.Matrix.at(i).at(j) = rhs.Tem[counter];
                    os << rhs.Matrix.at(i).at(j) << " | ";
                    counter = counter +1;
                }
                os << endl;
            }


            char winner;
            winner = rhs.win(rhs.Matrix);

            if (winner == 'X')
            {
                os << "\n";
                os << "Winner " << "X" << endl;
                os << "\n";
                os << "-----------------------------------------------" << endl;
            }
            else if(winner == 'O')
            {
                os << "\n";
                os << "Winner " << "O" << endl;
                os << "\n";
                os << "-----------------------------------------------" << endl;
            }

            else if (winner == 'M')
            {
                os << "\n";
                os << "Match Tie" << endl;
                os << "-----------------------------------------------" << endl;
            }

            else if(winner == 'A')
            {
                os << "\n";
                os << "X Turn " << endl;
                os << "\n";
                os << "Suggested Move by Computer  " << endl;

                vector <char> ComputedMove;
                os << "\n";

                ComputedMove = rhs.computer_move(rhs.Tem, 'X');

                int counter = 0 ;

                for (size_t i = 0; i < rhs.Matrix.size(); i++)
```



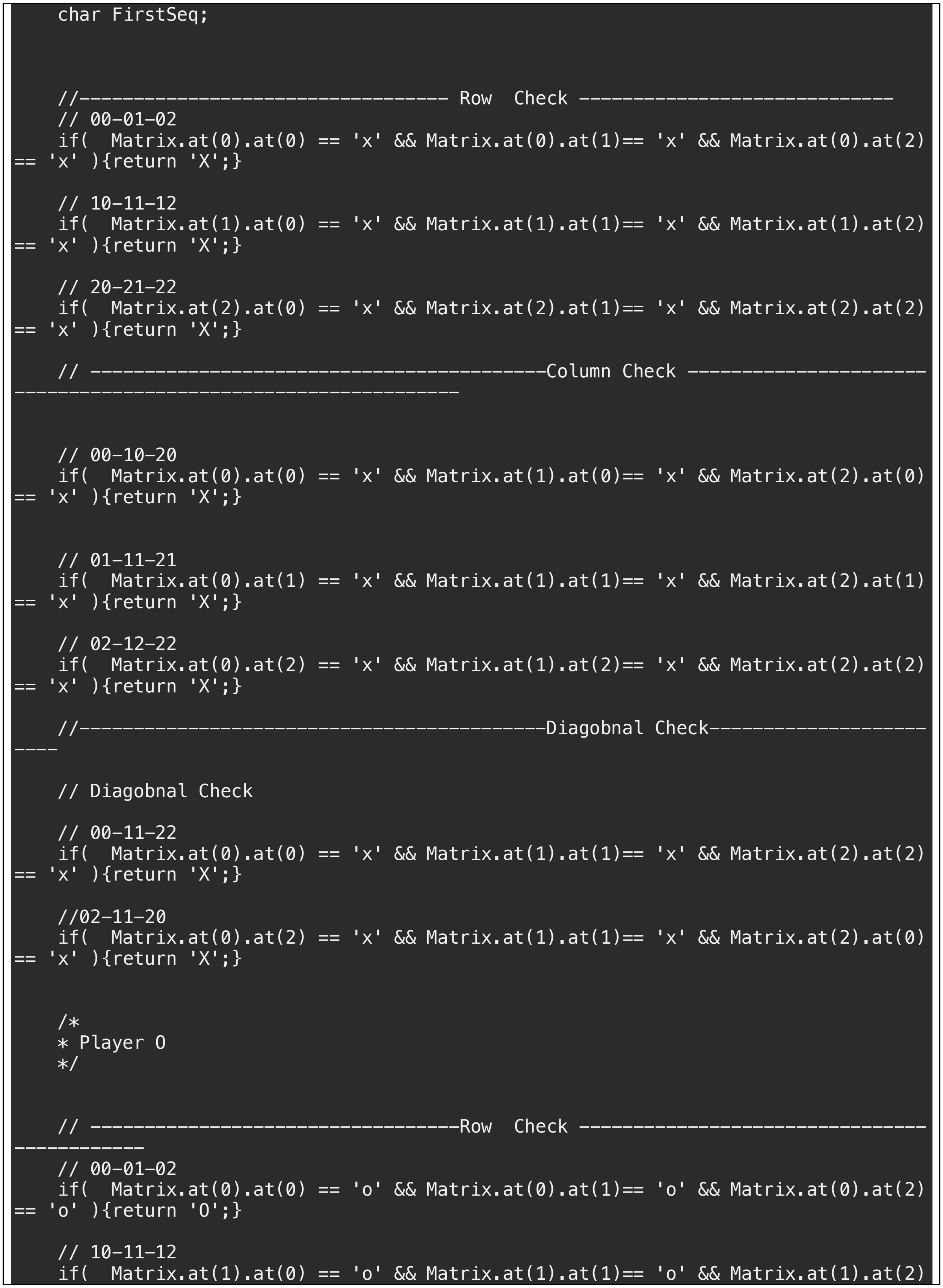
```
                {
                    for (size_t j = 0; j < rhs.Matrix.size(); j++)
                    {
                        rhs.Matrix.at(i).at(j) = ComputedMove[counter];
                        os << rhs.Matrix.at(i).at(j) << " | ";
                        counter = counter +1;

                    }
                    os << endl;
                }

                ComputedMove.clear();

                cout << "\n";
                os << "-----------------------------------------------" << endl;
            }

            else if(winner == 'B')
            {
                os << "\n";
                os << "O Turn " << endl;
                os << "\n";
                os << "Suggested Move by Computer  " << endl;

                vector <char> ComputedMove;
                os << "\n";

                ComputedMove = rhs.computer_move(rhs.Tem, 'O');

                int counter = 0 ;

                for (size_t i = 0; i < rhs.Matrix.size(); i++)
                {
                    for (size_t j = 0; j < rhs.Matrix.size(); j++)
                    {
                        rhs.Matrix.at(i).at(j) = ComputedMove[counter];
                        os << rhs.Matrix.at(i).at(j) << " | ";
                        counter = counter +1;

                    }
                    os << endl;
                }




                //rhs.print_matrix(ComputedMove);

                ComputedMove.clear();

                os << "-----------------------------------------------" << endl;
            }

        }

    return os;
}


vector <char> TicTacToe:: setPosition(string pos){
```



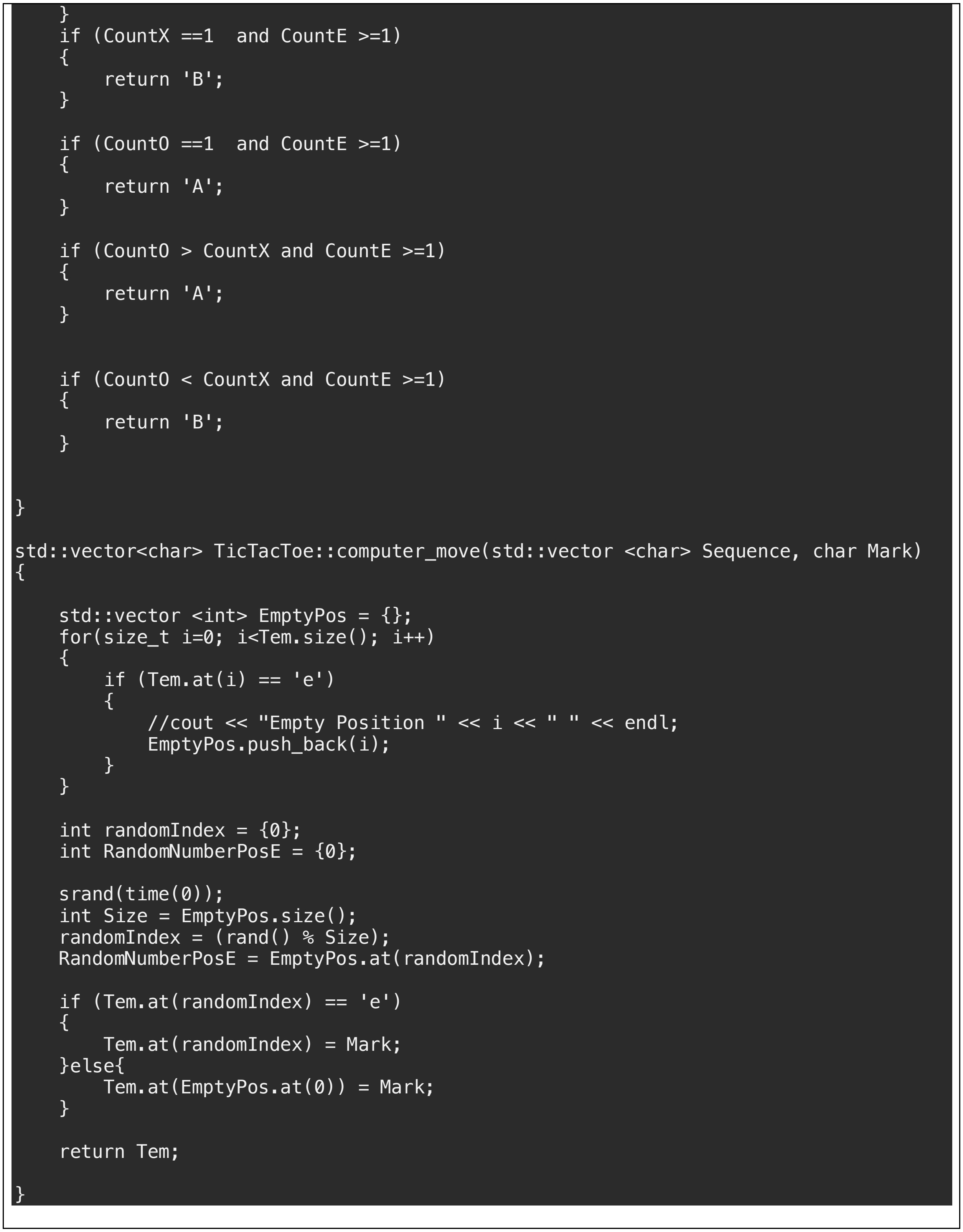
```
    Tem.clear();

    for(auto x:pos)
    {
        Tem.push_back(x);

    }
   return Tem;
}
TicTacToe::TicTacToe()
{
    cout << "\n" << "Welcome to Tic Tac Toe Game" << "\n" <<endl;

    for(size_t i=0 ; i<Matrix.size(); i ++)         // Row
    {
        for(size_t j=0 ; j<Matrix.size(); j ++)     //COL
        {
            cout << Matrix.at(i).at(j)<<" | ";      // Display on Console
        }
        cout << endl;
    }
    cout << "\n";
}

TicTacToe::~TicTacToe() {}

void TicTacToe:: print_matrix(std::vector <char> Sequence)
{
    /*
     * Takes a Vector type char Based on that it Print the Matrix
     */

    int counter = 0 ;
    for (size_t i = 0; i < Matrix.size(); i++)
    {

        for (size_t j = 0; j < Matrix.size(); j++) {

            Matrix.at(i).at(j) = Sequence[counter];
            cout << Matrix.at(i).at(j) << " | ";

            counter = counter +1;

        }
        cout << endl;

    }
    counter = 0;
}

char TicTacToe:: validate(std::vector <char> Sequence) const
{
    /*
     * This Function Return Boolean Based Whether the Input is Valid or Not
     */
    int GlobalCounter = 0;
    int CountX = 0;
    int CountO = 0;
    int CountE = 0;

    for(size_t i=0; i<Sequence.size(); i++)
```



```
    {

        if(Sequence.at(i) == 'x' or Sequence.at(i) == 'o' or Sequence.at(i) == 'e')
        {
            GlobalCounter = GlobalCounter + 1;

            if(Sequence.at(i) == 'x')
            {
                CountX = CountX + 1;
            }
            else if(Sequence.at(i) == 'o')
            {
                CountO = CountO + 1;
            }
            else if(Sequence.at(i) == 'e')
            {
                CountE = CountE + 1;
            }
        }
        else
        {
            return 'A';
            break;
        }
    }

    if (GlobalCounter != 9)
    {
        return 'B';
    }
    if(GlobalCounter == 9)
    {
        if (CountO >=6 or CountX >=6)
        {
            return 'C';
        }

        if(CountX > CountO and CountE ==1)
        {
            return 'C';
        }

        if(CountO > CountX and CountE ==1)
        {
            return 'C';
        }

    }

}

char TicTacToe::win(vector <vector <char>> Matrix){

    /*
     * Check if the Matrix Size if 3 so check for 3 Rows
     * Based on That check Condition for Winner
     */

    int CountX = 0;
    int CountO = 0;
    int CountE = 0;
```



```
    char FirstSeq;



    //---------------------------------- Row  Check -----------------------------
    // 00-01-02
    if(  Matrix.at(0).at(0) == 'x' && Matrix.at(0).at(1)== 'x' && Matrix.at(0).at(2)
== 'x' ){return 'X';}

    // 10-11-12
    if(  Matrix.at(1).at(0) == 'x' && Matrix.at(1).at(1)== 'x' && Matrix.at(1).at(2)
== 'x' ){return 'X';}

    // 20-21-22
    if(  Matrix.at(2).at(0) == 'x' && Matrix.at(2).at(1)== 'x' && Matrix.at(2).at(2)
== 'x' ){return 'X';}

    // ------------------------------------------Column Check ----------------------
-----------------------------------------


    // 00-10-20
    if(  Matrix.at(0).at(0) == 'x' && Matrix.at(1).at(0)== 'x' && Matrix.at(2).at(0)
== 'x' ){return 'X';}


    // 01-11-21
    if(  Matrix.at(0).at(1) == 'x' && Matrix.at(1).at(1)== 'x' && Matrix.at(2).at(1)
== 'x' ){return 'X';}

    // 02-12-22
    if(  Matrix.at(0).at(2) == 'x' && Matrix.at(1).at(2)== 'x' && Matrix.at(2).at(2)
== 'x' ){return 'X';}

    //-------------------------------------------Diagobnal Check--------------------
----

    // Diagobnal Check

    // 00-11-22
    if(  Matrix.at(0).at(0) == 'x' && Matrix.at(1).at(1)== 'x' && Matrix.at(2).at(2)
== 'x' ){return 'X';}

    //02-11-20
    if(  Matrix.at(0).at(2) == 'x' && Matrix.at(1).at(1)== 'x' && Matrix.at(2).at(0)
== 'x' ){return 'X';}


    /*
    * Player O
    */


    // ----------------------------------Row  Check --------------------------------
------------
    // 00-01-02
    if(  Matrix.at(0).at(0) == 'o' && Matrix.at(0).at(1)== 'o' && Matrix.at(0).at(2)
== 'o' ){return 'O';}

    // 10-11-12
    if(  Matrix.at(1).at(0) == 'o' && Matrix.at(1).at(1)== 'o' && Matrix.at(1).at(2)
== 'o' ){return 'O';}

    // 20-21-22
    if(  Matrix.at(2).at(0) == 'o' && Matrix.at(2).at(1)== 'o' && Matrix.at(2).at(2)
== 'o' ){return 'O';}


    // ------------------------------------------Column Check ----------------------
--------------

    // 00-10-20
    if(  Matrix.at(0).at(0) == 'o' && Matrix.at(1).at(0)== 'o' && Matrix.at(2).at(0)
== 'o' ){return 'O';}


    // 01-11-21
    if(  Matrix.at(0).at(1) == 'o' && Matrix.at(1).at(1)== 'o' && Matrix.at(2).at(1)
== 'o' ){return 'O';}

    // 02-12-22
    if(  Matrix.at(0).at(2) == 'o' && Matrix.at(1).at(2)== 'o' && Matrix.at(2).at(2)
== 'o' ){return 'O';}

    //------------------------------------------------------------------------------
----------------------------

    // Diagobnal Check

    // 00-11-22
    if(  Matrix.at(0).at(0) == 'o' && Matrix.at(1).at(1)== 'o' && Matrix.at(2).at(2)
== 'o' ){return 'O';}

    //02-11-20
    if(  Matrix.at(0).at(2) == 'o' && Matrix.at(1).at(1)== 'o' && Matrix.at(2).at(0)
== 'o' ){return 'O';}


    for(size_t i=0; i < Matrix.size(); i ++)
    {
        for(size_t j=0; j<Matrix.size(); j++){
            if(Matrix.at(i).at(j) == 'x'){
                CountX = CountX+1;
            }
            if(Matrix.at(i).at(j) == 'o'){
                CountO = CountO +1;
            }

            if(Matrix.at(i).at(j)== 'e'){
                CountE = CountE+1;
            }

        }
    }

    if (CountE == 0)
    {
        return 'M';
    }
    if(CountX ==  CountO and CountE == 1)
    {
        return 'A';
```

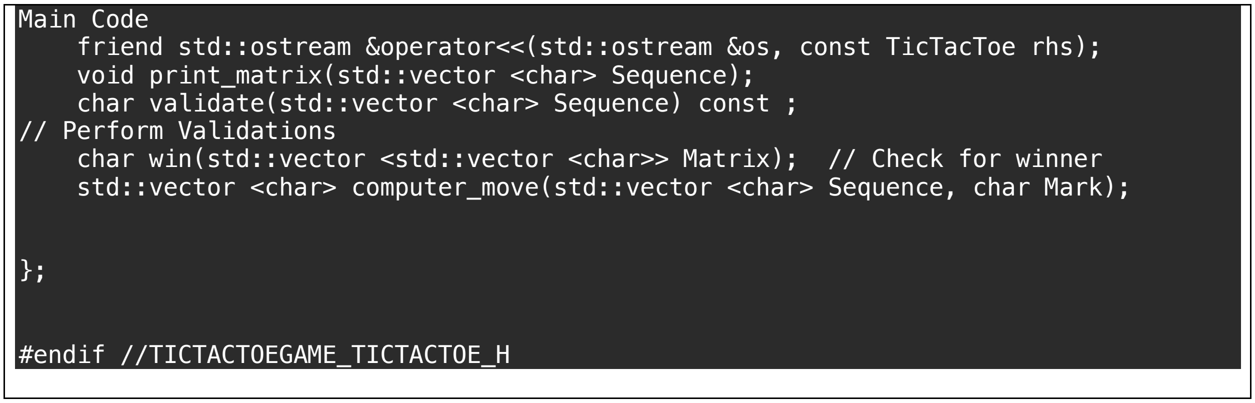


```
    }
    if (CountX ==1  and CountE >=1)
    {
        return 'B';
    }

    if (CountO ==1  and CountE >=1)
    {
        return 'A';
    }

    if (CountO > CountX and CountE >=1)
    {
        return 'A';
    }


    if (CountO < CountX and CountE >=1)
    {
        return 'B';
    }


}

std::vector<char> TicTacToe::computer_move(std::vector <char> Sequence, char Mark)
{

    std::vector <int> EmptyPos = {};
    for(size_t i=0; i<Tem.size(); i++)
    {
        if (Tem.at(i) == 'e')
        {
            //cout << "Empty Position " << i << " " << endl;
            EmptyPos.push_back(i);
        }
    }

    int randomIndex = {0};
    int RandomNumberPosE = {0};

    srand(time(0));
    int Size = EmptyPos.size();
    randomIndex = (rand() % Size);
    RandomNumberPosE = EmptyPos.at(randomIndex);

    if (Tem.at(randomIndex) == 'e')
    {
        Tem.at(randomIndex) = Mark;
    }else{
        Tem.at(EmptyPos.at(0)) = Mark;
    }

    return Tem;

}
```

Source Code:

TicTacToe.h

```
/*
 * TicTacToe.h File
 * Tic Tac Toe Game
 * Author : Soumil Nitin SHah
 * Version 0.0.1
 * Email soushah@my.bridgeport.edu
 * Student id : 1031686
 * Github:
 * Last Modified : 14 Sepetember 2019
 *
 * Operating System: MAC os
 * Configured with: --prefix=/Applications/Xcode.app/Contents/Developer/usr --with-
gxx-include-
dir=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/S
DKs/MacOSX10.14.sdk/usr/include/c++/4.2.
 * Apple LLVM version 10.0.1 (clang-1001.0.46.3)
 * Target: x86_64-apple-darwin18.7.0
 * InstalledDir:
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/b
in
 */

#ifndef TICTACTOEGAME_TICTACTOE_H
#define TICTACTOEGAME_TICTACTOE_H

#include "TicTacToe.h"
#include <fstream>
#include <vector>
#include <string>
#include <iostream>
#include <ostream>


class TicTacToe {

public:
    /*
     * Declaring Variables
     */
    std::vector <std::vector <char>> Matrix {
            {'e','e','e'},
            {'e','e','e'},
            {'e','e','e'}
    };
    std::vector <char> Tem;

public:
    /*
     * Constructor and Destructor
     */
    TicTacToe();                                                              //
Constructor
    ~TicTacToe();                                                             //
Destructor

    std::vector <char> setPosition(std::string pos);                          //
```



```
Main Code
    friend std::ostream &operator<<(std::ostream &os, const TicTacToe rhs);
    void print_matrix(std::vector <char> Sequence);
    char validate(std::vector <char> Sequence) const ;
// Perform Validations
    char win(std::vector <std::vector <char>> Matrix);  // Check for winner
    std::vector <char> computer_move(std::vector <char> Sequence, char Mark);


};


#endif //TICTACTOEGAME_TICTACTOE_H
```