# LAB PROJECT

Fundamental of programming

**GROUP MEMBERS:**

| | |
|---|---|
| 01: TALHA SHAH | 461075 |
| 02: SAMEEN WASEEM | 465064 |
| 03: AATIKA KAMRAN | 464185 |

**SUBMITTED TO:**

SIR AFFAN

DECEMBER 20, 2023

NUST,SCHOOL OF MECHANICAL AND MANUFACTURING ENGINEERING
H-12 ISLAMABAD

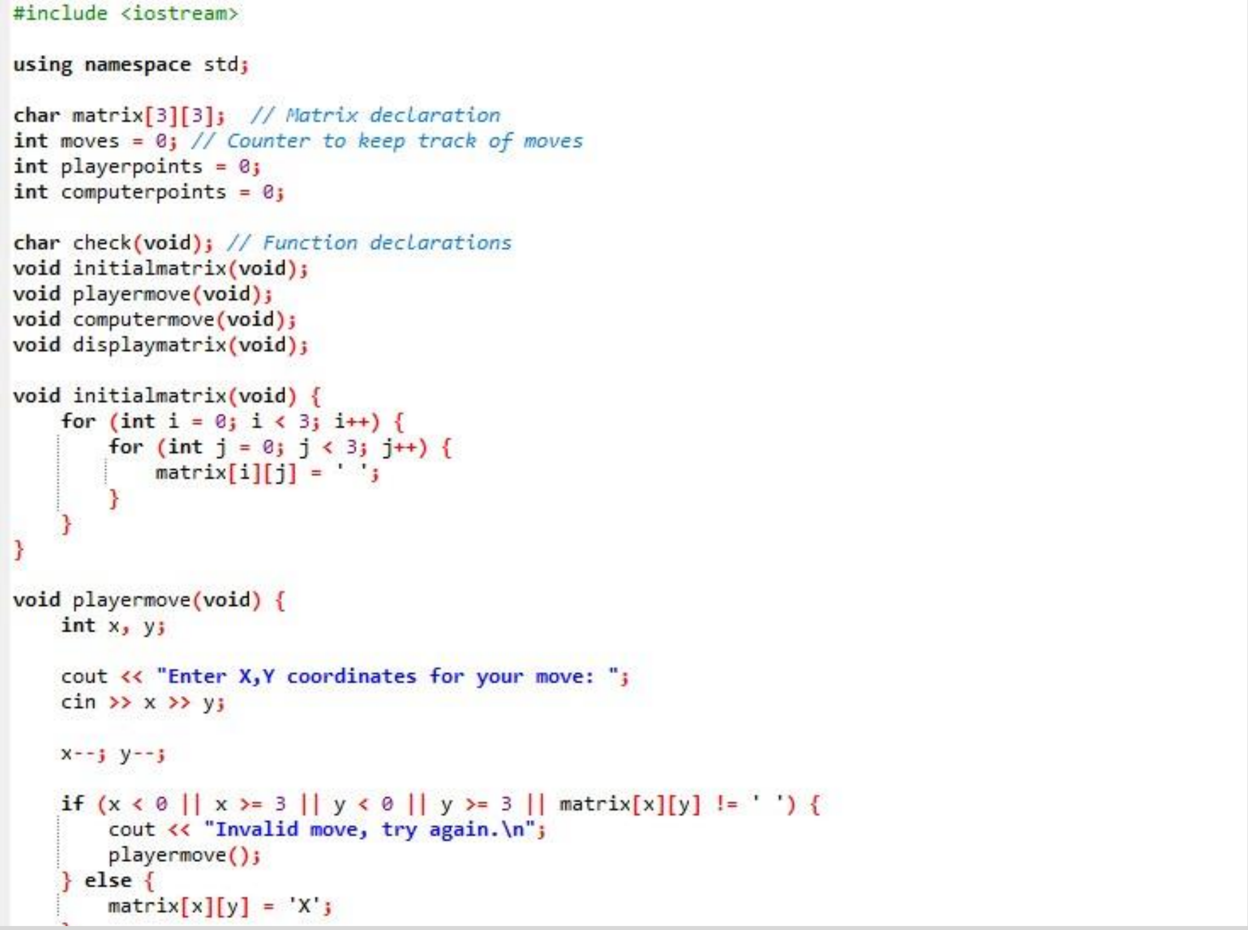
## INTRODUCTION:

The following  C++code is of small game named as tic tac toe played by two player one

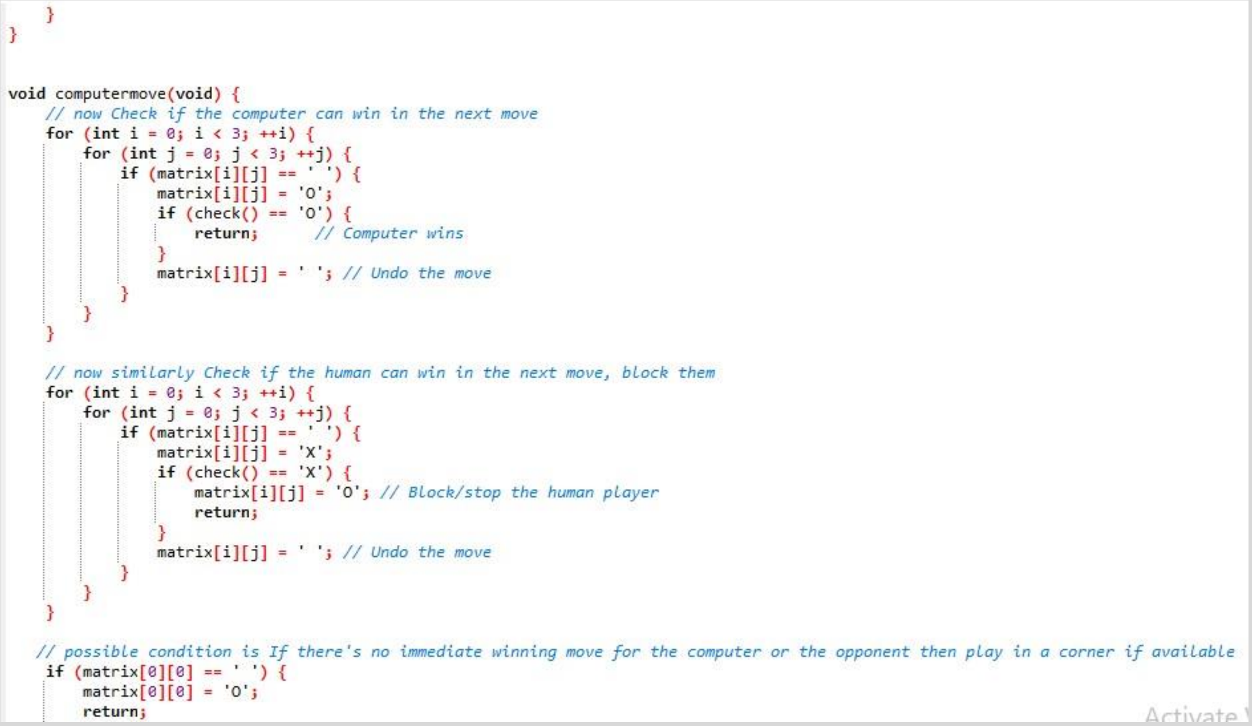
Human(user) and other AI(computer).

In this game first user paly its move and in response of this AI plays its move. We made three round of game so that at the end point calculated to give final results.

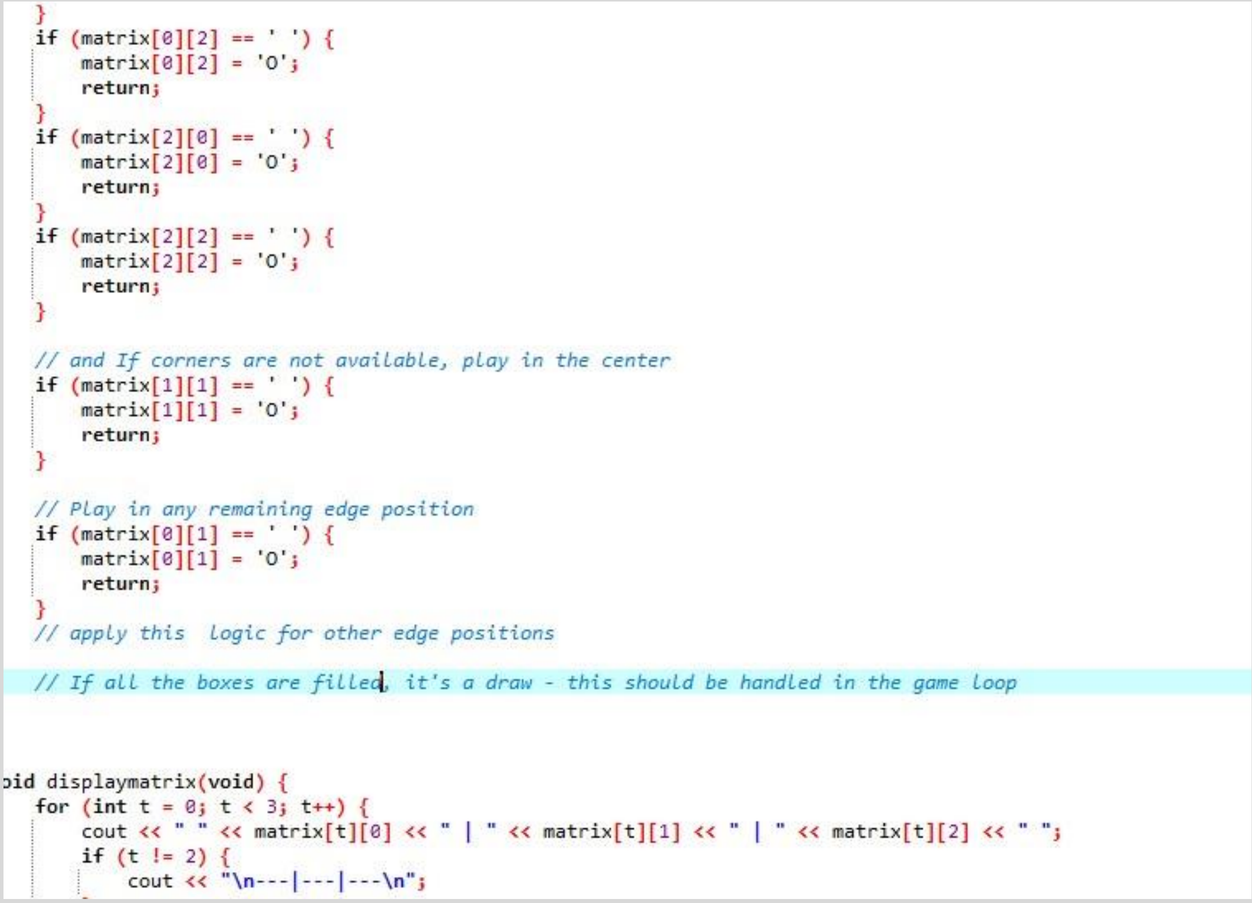
## CODE:



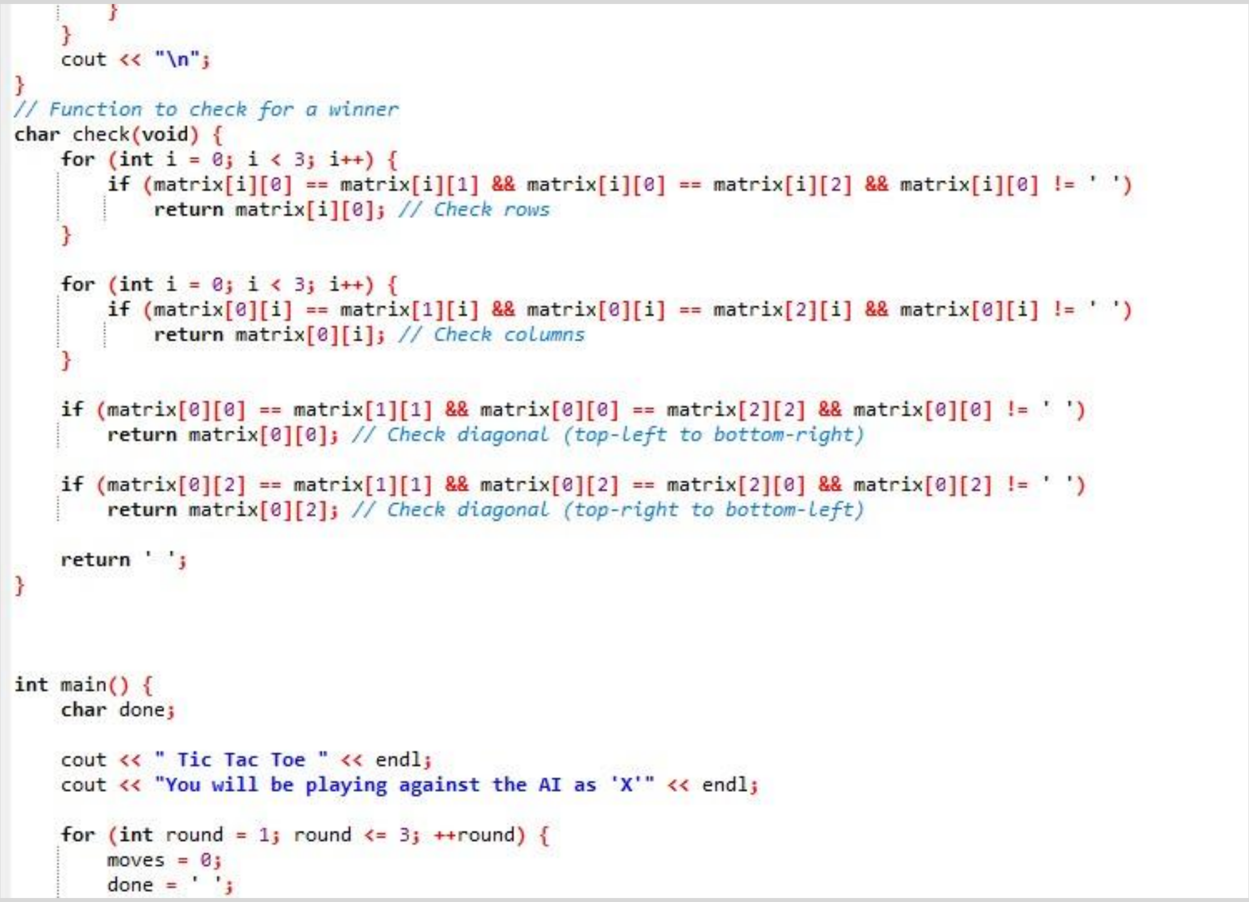
```
#include <iostream>

using namespace std;

char matrix[3][3];  // Matrix declaration
int moves = 0; // Counter to keep track of moves
int playerpoints = 0;
int computerpoints = 0;

char check(void); // Function declarations
void initialmatrix(void);
void playermove(void);
void computermove(void);
void displaymatrix(void);

void initialmatrix(void) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            matrix[i][j] = ' ';
        }
    }
}

void playermove(void) {
    int x, y;

    cout << "Enter X,Y coordinates for your move: ";
    cin >> x >> y;

    x--; y--;

    if (x < 0 || x >= 3 || y < 0 || y >= 3 || matrix[x][y] != ' ') {
        cout << "Invalid move, try again.\n";
        playermove();
    } else {
        matrix[x][y] = 'X';
```



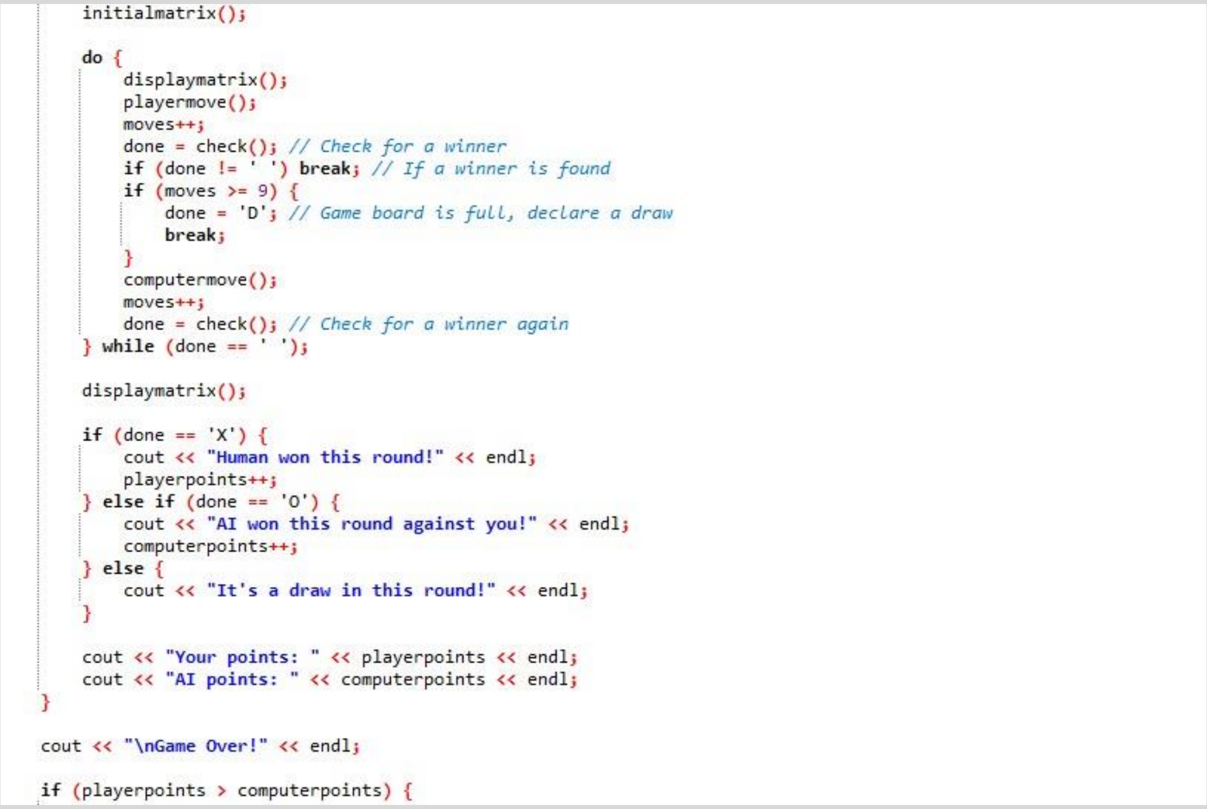
```
    }
}


void computermove(void) {
    // now Check if the computer can win in the next move
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            if (matrix[i][j] == ' ') {
                matrix[i][j] = 'O';
                if (check() == 'O') {
                    return;      // Computer wins
                }
                matrix[i][j] = ' '; // Undo the move
            }
        }
    }

    // now similarly Check if the human can win in the next move, block them
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            if (matrix[i][j] == ' ') {
                matrix[i][j] = 'X';
                if (check() == 'X') {
                    matrix[i][j] = 'O'; // Block/stop the human player
                    return;
                }
                matrix[i][j] = ' '; // Undo the move
            }
        }
    }

   // possible condition is If there's no immediate winning move for the computer or the opponent then play in a corner if available
    if (matrix[0][0] == ' ') {
        matrix[0][0] = 'O';
        return;
```



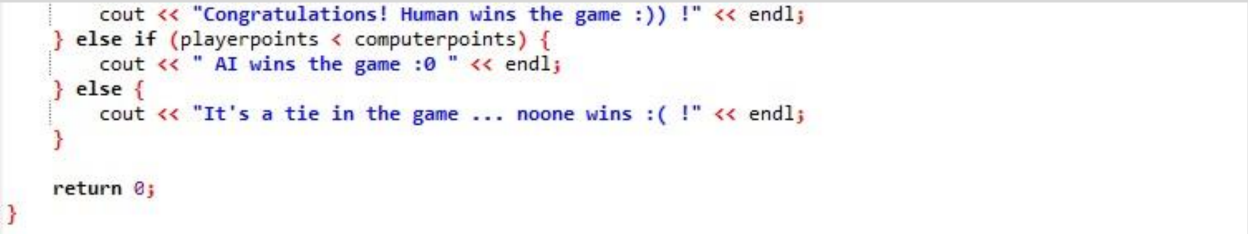
```
    }
    if (matrix[0][2] == ' ') {
        matrix[0][2] = 'O';
        return;
    }
    if (matrix[2][0] == ' ') {
        matrix[2][0] = 'O';
        return;
    }
    if (matrix[2][2] == ' ') {
        matrix[2][2] = 'O';
        return;
    }

    // and If corners are not available, play in the center
    if (matrix[1][1] == ' ') {
        matrix[1][1] = 'O';
        return;
    }

    // Play in any remaining edge position
    if (matrix[0][1] == ' ') {
        matrix[0][1] = 'O';
        return;
    }
    // apply this  logic for other edge positions

    // If all the boxes are filled, it's a draw - this should be handled in the game loop


oid displaymatrix(void) {
    for (int t = 0; t < 3; t++) {
        cout << " " << matrix[t][0] << " | " << matrix[t][1] << " | " << matrix[t][2] << " ";
        if (t != 2) {
            cout << "\n---|---|---\n";
```



```
        }
    }
    cout << "\n";
}
// Function to check for a winner
char check(void) {
    for (int i = 0; i < 3; i++) {
        if (matrix[i][0] == matrix[i][1] && matrix[i][0] == matrix[i][2] && matrix[i][0] != ' ')
            return matrix[i][0]; // Check rows
    }

    for (int i = 0; i < 3; i++) {
        if (matrix[0][i] == matrix[1][i] && matrix[0][i] == matrix[2][i] && matrix[0][i] != ' ')
            return matrix[0][i]; // Check columns
    }

    if (matrix[0][0] == matrix[1][1] && matrix[0][0] == matrix[2][2] && matrix[0][0] != ' ')
        return matrix[0][0]; // Check diagonal (top-left to bottom-right)

    if (matrix[0][2] == matrix[1][1] && matrix[0][2] == matrix[2][0] && matrix[0][2] != ' ')
        return matrix[0][2]; // Check diagonal (top-right to bottom-left)

    return ' ';
}



int main() {
    char done;

    cout << " Tic Tac Toe " << endl;
    cout << "You will be playing against the AI as 'X'" << endl;

    for (int round = 1; round <= 3; ++round) {
        moves = 0;
        done = ' ';
```



```
        initialmatrix();

        do {
            displaymatrix();
            playermove();
            moves++;
            done = check(); // Check for a winner
            if (done != ' ') break; // If a winner is found
            if (moves >= 9) {
                done = 'D'; // Game board is full, declare a draw
                break;
            }
            computermove();
            moves++;
            done = check(); // Check for a winner again
        } while (done == ' ');

        displaymatrix();

        if (done == 'X') {
            cout << "Human won this round!" << endl;
            playerpoints++;
        } else if (done == 'O') {
            cout << "AI won this round against you!" << endl;
            computerpoints++;
        } else {
            cout << "It's a draw in this round!" << endl;
        }

        cout << "Your points: " << playerpoints << endl;
        cout << "AI points: " << computerpoints << endl;
    }

    cout << "\nGame Over!" << endl;

    if (playerpoints > computerpoints) {
```



```
        cout << "Congratulations! Human wins the game :)) !" << endl;
    } else if (playerpoints < computerpoints) {
        cout << " AI wins the game :0 " << endl;
    } else {
        cout << "It's a tie in the game ... noone wins :( !" << endl;
    }

    return 0;
}
```

## **CODE EXPLANTION:**

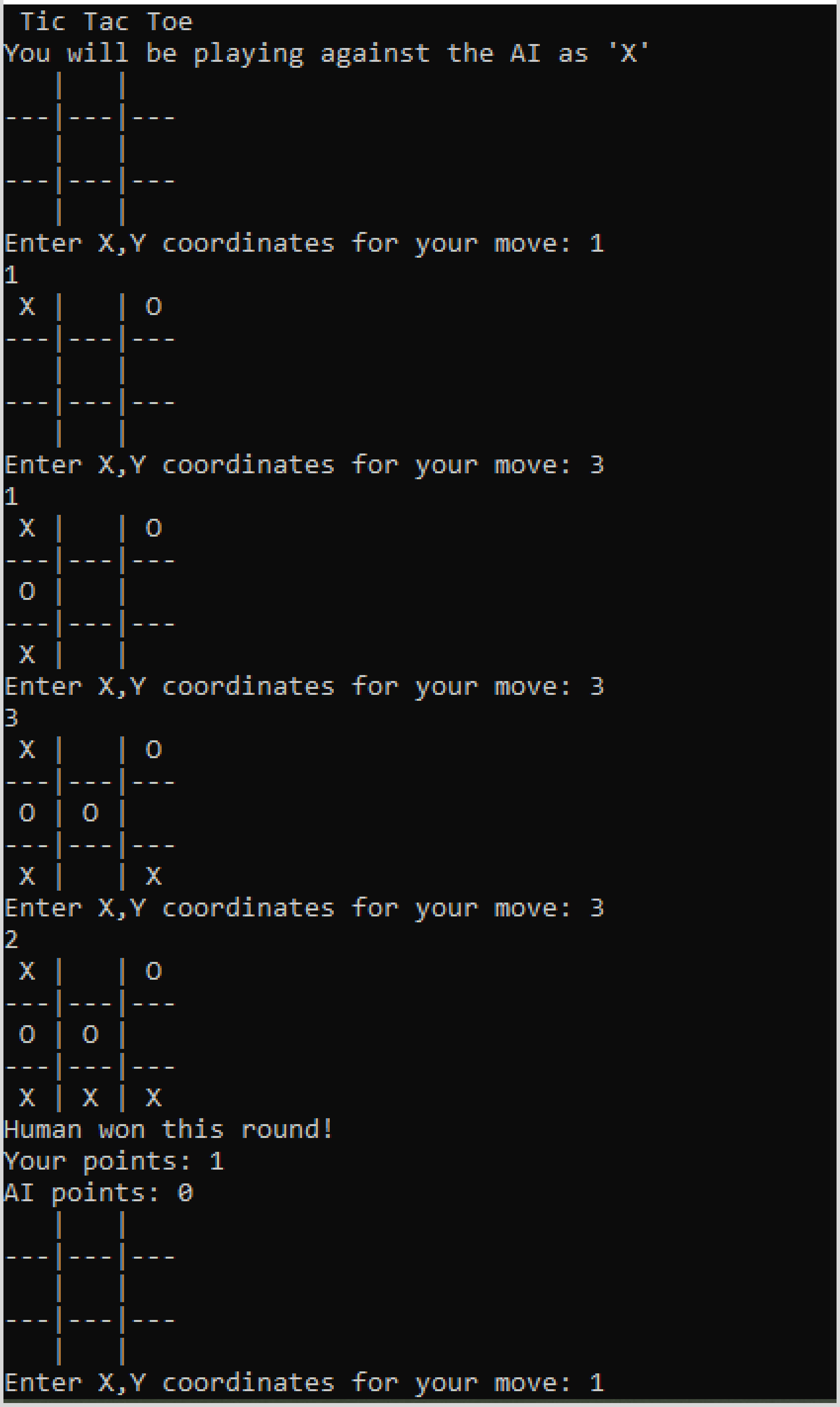
In our code we use all the concept which we have studied in the class.We use data type of integers characters and we use a void functions in which we do not return our value.We made a matrix of 3x3 using array and store values in it. For making a game board we first simply store spaces in matrix.

We then set up player move in the form of coordinates of matrix 3x3 in erturn to this AI play it moves here we use if else statement.

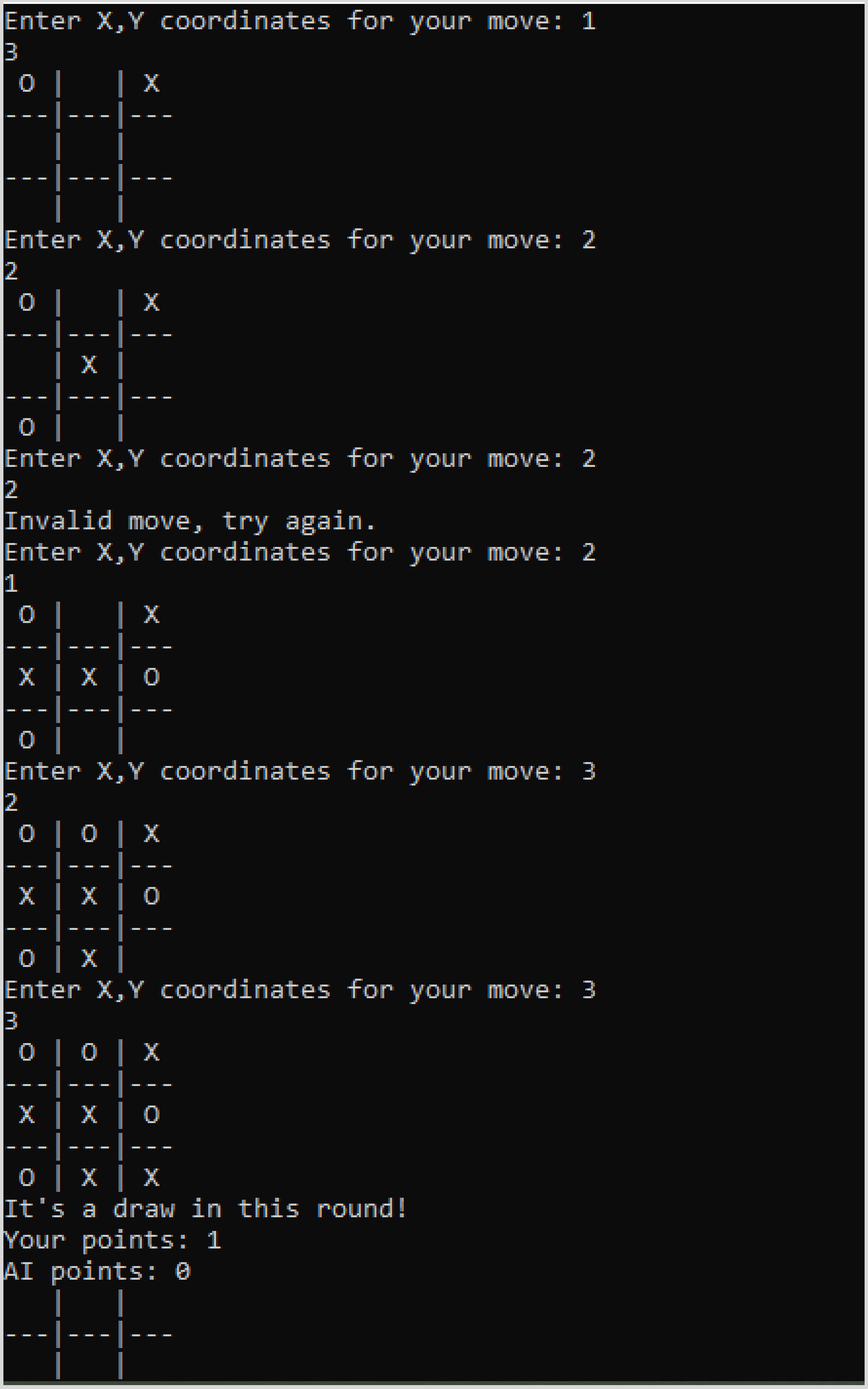
Using loop and nested loop for playing moves both AI and user. All the possible condition for winning is enter so according to that moves should be carried out.

Our game is of three rounds which player get more point that one will win. After all moves are played then check function is performed.

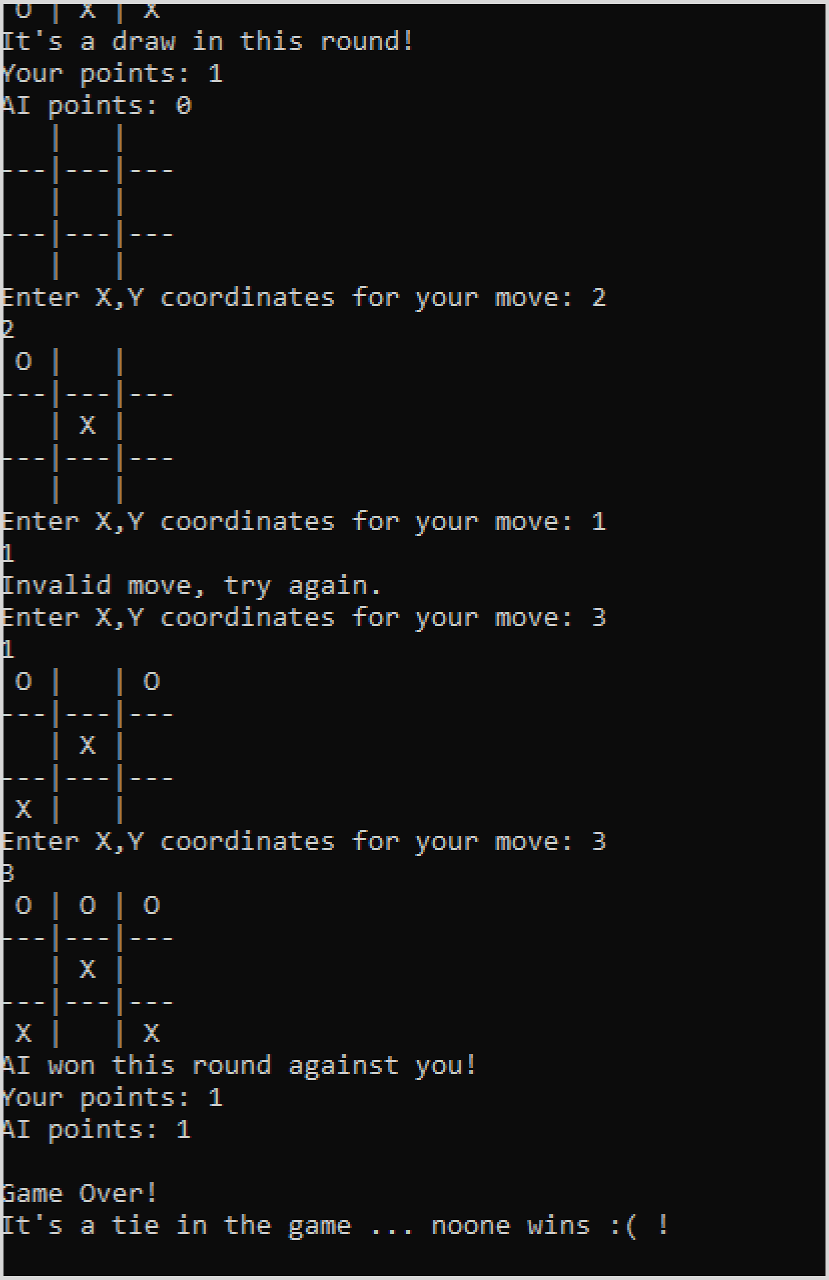
## **OUTPUT:**



```
 Tic Tac Toe
You will be playing against the AI as 'X'
   |   |
---|---|---
   |   |
---|---|---
   |   |
Enter X,Y coordinates for your move: 1
1
 X |   | O
---|---|---
   |   |
---|---|---
   |   |
Enter X,Y coordinates for your move: 3
1
 X |   | O
---|---|---
 O |   |
---|---|---
 X |   |
Enter X,Y coordinates for your move: 3
3
 X |   | O
---|---|---
 O | O |
---|---|---
 X |   | X
Enter X,Y coordinates for your move: 3
2
 X |   | O
---|---|---
 O | O |
---|---|---
 X | X | X
Human won this round!
Your points: 1
AI points: 0
   |   |
---|---|---
   |   |
---|---|---
   |   |
Enter X,Y coordinates for your move: 1
```



```
Enter X,Y coordinates for your move: 1
3
 O |   | X
---|---|---
   |   |
---|---|---
   |   |
Enter X,Y coordinates for your move: 2
2
 O |   | X
---|---|---
   | X |
---|---|---
 O |   |
Enter X,Y coordinates for your move: 2
2
Invalid move, try again.
Enter X,Y coordinates for your move: 2
1
 O |   | X
---|---|---
 X | X | O
---|---|---
 O |   |
Enter X,Y coordinates for your move: 3
2
 O | O | X
---|---|---
 X | X | O
---|---|---
 O | X |
Enter X,Y coordinates for your move: 3
3
 O | O | X
---|---|---
 X | X | O
---|---|---
 O | X | X
It's a draw in this round!
Your points: 1
AI points: 0
   |   |
---|---|---
   |   |
```



```
 O | X | X
It's a draw in this round!
Your points: 1
AI points: 0
   |   |
---|---|---
   |   |
---|---|---
   |   |
Enter X,Y coordinates for your move: 2
2
 O |   |
---|---|---
   | X |
---|---|---
   |   |
Enter X,Y coordinates for your move: 1
1
Invalid move, try again.
Enter X,Y coordinates for your move: 3
1
 O |   | O
---|---|---
   | X |
---|---|---
 X |   |
Enter X,Y coordinates for your move: 3
3
 O | O | O
---|---|---
   | X |
---|---|---
 X |   | X
AI won this round against you!
Your points: 1
AI points: 1

Game Over!
It's a tie in the game ... noone wins :( !
```

## **OUTPUT EXPLANATION:**

Output of code is shown above is simple we simple enter coordinates of matrix for our moves and after all the blocks are filled then it gives you result either user win or AI wins or match is draw.

## **CONCLUSIONS:**

Hence the following code is useful and function used by it are simple our game give us three possibilities winner, losser and draw match.