```cpp
#include<iostream>
using namespace std;

bool win_game_row(char arr[3][3]){
    //Checks Horizontal Win
    for(int i=0;i<3;i++){
        bool res = true;
        char same = arr[i][0];
        if(same == '-'){
            continue; //This Row has empty spaces
        }
        for(int j=1;j<3;j++){
            if(arr[i][j] != same){
                res = false;
                break;
            }
        }
        if(res == true){
            return res;
        }
    }
    return false;
}

bool win_game_col(char arr[3][3]){
    //Checks Vertical Win
    for(int i=0;i<3;i++){
        bool res = true;
        char same = arr[0][i];
        if(same == '-'){
            continue; //This Column has empty spaces
        }
        for(int j=1;j<3;j++){
            if(arr[j][i] != same){
                res = false;
                break;
            }
        }
        if(res == true){
            return res;
        }
    }
    return false;
}

bool win_game_diag1(char arr[3][3]){
    //Checks Diagonal Win
    char same = arr[0][0];
    if(same == '-'){
        return false;
    }
    for(int i=1;i<3;i++){
        if(arr[i][i] != same){
            return false;
        }
    }
```

```
56:     }
57:     return true;
58: }
59:
60: bool win_game_diag2(char arr[3][3]){
61:     //Checks Diagonal Win
62:     char same = arr[0][2];
63:     if(same == '-'){
64:         return false;
65:     }
66:     for(int i=1;i<3;i++){
67:         if(arr[i][2-i] != same){
68:             return false;
69:         }
70:     }
71:     return true;
72: }
73:
74: void display(char arr[3][3]){
75:     //Displays the complete board
76:     cout<<"\n";
77:     cout<<arr[0][0]<<" | "<<arr[0][1]<<" | "<<arr[0][2]<<endl;
78:     cout<<"---------\n";
79:     cout<<arr[1][0]<<" | "<<arr[1][1]<<" | "<<arr[1][2]<<endl;
80:     cout<<"---------\n";
81:     cout<<arr[2][0]<<" | "<<arr[2][1]<<" | "<<arr[2][2]<<endl;
82: }
83:
84: bool completelyFilled(char arr[3][3]){
85:     bool res = true;
86:     for(int i=0;i<3;i++){
87:         for(int j=0;j<3;j++){
88:             if(arr[i][j] == '-'){
89:                 res = false;
90:                 return res;
91:             }
92:         }
93:     }
94:     return res;
95: }
96:
97: void change(char arr[3][3],int row,int col,int turn){
98:     //Takes input from the user and changes the value at that position
99:     if(turn == 0){
100:         arr[row][col] = 'O';
101:     }
102:     else{
103:         arr[row][col] = 'X';
104:     }
105: }
106:
107: bool filled(char arr[3][3],int row,int col){
108:     //Check whether the given position is already occupied or not
109:     return (arr[row][col] == 'X' || arr[row][col] == 'O');
110: }
```

```
111:
112: int main(){
113:     begin:
114:         char arr[3][3];
115:         fill_n(arr[0],3,'-');
116:         fill_n(arr[1],3,'-');
117:         fill_n(arr[2],3,'-');
118:         int turn = 1;
119:         string temp_turn = "a"; //Temporary variable to read input of turn from the user
120:         //Creating this will enable us to read all the characters without any segmentation fa
121:
122:         cout<<"Player-1 will play with \'X\' \n"; //turn = 1
123:         cout<<"Player-2 will play with \'O\' \n"; //turn = 0
124:         cout<<"\n";
125:         while(true){
126:             cout<<"Which Player will turn first (1 or 2): ";
127:             getline(cin,temp_turn);
128:
129:             if(temp_turn == "1"){
130:                 turn = 1; //0 represents Player-1 who plays with X
131:                 break;
132:             }
133:             else if(temp_turn == "2"){
134:                 turn = 0; //0 represents Player-2 who plays with O
135:                 break;
136:             }
137:             else{
138:                 cout<<"Wrong Input! Try Again\n\n";
139:             }
140:         }
141:
142:         int row,col;
143:         while(true){
144:             while(true){ //Checking whether the place is already filled or not
145:                 cout<<"\nEnter a row number (1 or 2 or 3): ";
146:                 cin>>row;
147:                 cout<<"Enter a column number (1 or 2 or 3): ";
148:                 cin>>col;
149:                 if(row<=0 || row>=4){
150:                     cout<<"Invalid Row Number Input! Try Again\n";
151:                 }
152:                 else if(col<=0 || col>=4){
153:                     cout<<"Invalid Column Number Input! Try Again\n";
154:                 }
155:                 else if(filled(arr,row-1,col-1)){
156:                     cout<<"Place Already Filled! Try Again\n";
157:                 }
158:                 else{
159:                     break;
160:                 }
161:             }
162:             change(arr,row-1,col-1,turn); //Passing Player Input to TIC-TAC-TOE Board
163:             display(arr); //Displaying TIC-TAC-TOE Board
164:             if(win_game_col(arr) || win_game_row(arr) || win_game_diag1(arr) || win_game_diag
165:                 cout<<(turn==0?"Player-2 won the game!\n":"Player-1 won the game!\n");
```

```
166:                 break;
167:             }
168:             if(completelyFilled(arr)){
169:                 cout<<"\n****It's a tie!!****\n";
170:                 cout<<"Board has been completely filled\n";
171:                 break;
172:             }
173:             turn = (turn+1)%2;
174:         }
175:         string choice;
176:         cout<<"\n\nDo you want to play again? (y/n) ";
177:         cout<<"\nPress any other key to exit: ";
178:         getline(cin,choice);
179:         getline(cin,choice);
180:         if(choice=="y"){
181:             cout<<endl;
182:             goto begin;
183:         }
184: }
```