Tic-Tac-Toe (with Minimax AI)

This project implements a Tic-Tac-Toe game in C++ where a human can play against the computer. The computer uses the Minimax algorithm to play optimally, making it impossible to beat if played correctly.

Features:

- Human vs Computer gameplay.
- Computer plays optimally using Minimax algorithm.
- Board visualization in the terminal.
- Instructions for easy cell selection.
- Detects wins, losses, and draws.
- Input validation (invalid positions / already occupied cells).
- Option to choose whether human or computer starts first.
- Play multiple games in one run.

Technologies Used:

```
- C++ STL (<bits/stdc++.h>, loops, conditionals)
- Algorithms:
  * Minimax (recursively evaluates moves for best outcome)
  * Game-over detection for rows, columns, and diagonals
```

File Structure:

```
.
■■■ Tic_tac_toe.cpp    # Source code for Tic-Tac-Toe
■■■ TicTacToe_README.pdf   # Documentation
```

How to Compile and Run:

```
Compile:
g++ Tic_tac_toe.cpp -o tictactoe

Run:
./tictactoe
```

Gameplay Instructions:

```
The board cells are numbered 1-9 as follows:

 1 | 2 | 3
-----------
 4 | 5 | 6
-----------
 7 | 8 | 9

- Choose a cell number to place your X.
- The computer places O automatically in the optimal position.
- The game continues until:
  * Human wins
  * Computer wins
  * Or the game is a draw
```

Example Gameplay:

```
------------------------------------------------------------------
                          Tic-Tac-Toe
------------------------------------------------------------------

Do you want to start first?(y/n) : y

Choose a cell numbered from 1 to 9 as below and play

              1 | 2 | 3
             -----------
              4 | 5 | 6
             -----------
              7 | 8 | 9

You can insert in the following positions : 1 2 3 4 5 6 7 8 9
Enter the position = 5

HUMAN has put a X in cell 5

                |   |
             -----------
                | X |
             -----------
                |   |
```

Customization:

```
- Change symbols by modifying:
  #define COMPUTERMOVE 'O'
  #define HUMANMOVE 'X'

- Adjust board size by changing:
  #define SIDE 3
(Current code supports 3×3 only.)
```

Future Improvements:

- Extend to NxN Tic-Tac-Toe.

- Add difficulty levels (easy/medium/hard).

- GUI version with graphics.