```
//TIC TAC TOE GAME

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <time.h>

char board[3][3];
const char PLAYER = 'X';
const char COMPUTER = 'O';

void resetBoard();
void printBoard();
int checkFreeSpaces();
void playerMove();
void computerMove();
char checkWinner();
void printWinner(char);

int main()
{
  char winner = ' ';
  char response = ' ';

    winner = ' ';
    response = ' ';
    resetBoard();

    while(winner == ' ' && checkFreeSpaces() != 0)
    {
      printBoard();
```

```
        playerMove();
        winner = checkWinner();
        if(winner != ' ' || checkFreeSpaces() == 0)
        {
          break;
        }

        computerMove();
        winner = checkWinner();
        if(winner != ' ' || checkFreeSpaces() == 0)
        {
          break;
        }
      }

      printBoard();
      printWinner(winner);

   printf("Thanks for playing!");
 return 0;
}

void resetBoard()
{
   for(int i = 0; i < 3; i++)
   {
      for(int j = 0; j < 3; j++)
      {
        board[i][j] = ' ';
      }
```

```
  }
}
void printBoard()
{
  printf(" %c | %c | %c ", board[0][0], board[0][1], board[0][2]);
  printf("\n---|---|---\n");
  printf(" %c | %c | %c ", board[1][0], board[1][1], board[1][2]);
  printf("\n---|---|---\n");
  printf(" %c | %c | %c ", board[2][0], board[2][1], board[2][2]);
  printf("\n");
}
int checkFreeSpaces()
{
  int freeSpaces = 9;

  for(int i = 0; i < 3; i++)
  {
    for(int j = 0; j < 3; j++)
    {
      if(board[i][j] != ' ')
      {
        freeSpaces--;
      }
    }
  }
  return freeSpaces;
}
void playerMove()
{
  int x;
  int y;
```

```
  do
  {
    printf("Enter row #(1-3): ");
    scanf("%d", &x);
    x--;
    printf("Enter column #(1-3): ");
    scanf("%d", &y);
    y--;

    if(board[x][y] != ' ')
    {
      printf("Invalid move!\n");
    }
    else
    {
      board[x][y] = PLAYER;
      break;
    }
  } while (board[x][y] != ' ');

}
void computerMove()
{
  //will create a seed based on current time
  srand(time(0));
  int x;
  int y;

  if(checkFreeSpaces() > 0)
  {
```

```
    do
    {
      x = rand() % 3;
      y = rand() % 3;
    } while (board[x][y] != ' ');

    board[x][y] = COMPUTER;
  }
  else
  {
    printWinner(' ');
  }
}
char checkWinner()
{
  //To check rows
  for(int i = 0; i < 3; i++)
  {
    if(board[i][0] == board[i][1] && board[i][0] == board[i][2])
    {
      return board[i][0];
    }
  }
  //To check columns
  for(int i = 0; i < 3; i++)
  {
    if(board[0][i] == board[1][i] && board[0][i] == board[2][i])
    {
      return board[0][i];
    }
  }
```

```
  //To check diagonals
  if(board[0][0] == board[1][1] && board[0][0] == board[2][2])
  {
    return board[0][0];
  }
  if(board[0][2] == board[1][1] && board[0][2] == board[2][0])
  {
    return board[0][2];
  }

  return ' ';
}
void printWinner(char winner)
{
  if(winner == PLAYER)
  {
    printf("YOU WIN! :)");
  }
  else if(winner == COMPUTER)
  {
    printf("YOU LOSE! :(");
  }
  else{
    printf("IT'S A TIE! :|");
  }
}
```