TIC-TOC-TOE

**WHAT IS TIC-TOC-TOE?**

Tic-tac-toe is not a very challenging game for human beings. If you’re an enthusiast, you’ve probably moved from the basic game to some variant like threedimensional tic-tac-toe on a larger grid. If you sit downright now to play ordinary three-by-three tic-tac-toe with a friend, what will probably happen is that every game will come out a tie. Both you and your friend can probably play perfectly, never making a mistake that would allow your opponent to win. But can you describe how you know where to move each turn? Most of the time,

you probably aren’t even aware of alternative

 possibilities; you just look at the board and instantly know where you want to move. That kind of instant knowledge is great for human beings, because it makes you a fast

 player. But it isn’t much help in writing a

 computer program. For that, you have to know very explicitly what your strategy is.

**REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| ID | DESCRIPTION | STATUS |

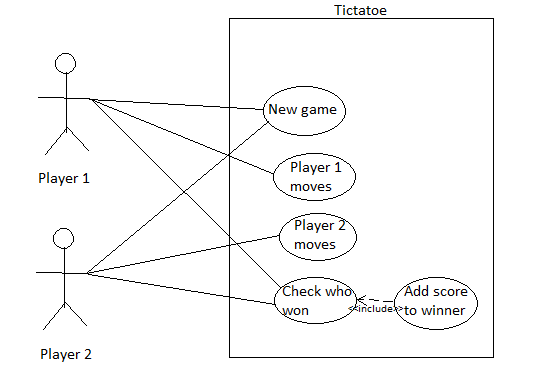
* LOW LEVEL REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| LLR01 | 9 SQUARE | IMPLEMENTED |
| LLR02 | PLAYER 1 | IMPLEMENTED |
| LLR03 | PLAYER 2 | IMPLEMENTED |

* HIGH LEVEL REQUIREMENTS

|  |  |  |
| --- | --- | --- |
| ID | DESCRIPTION | STATUS |
| HLR01 | C PROGRAMMING | IMPLEMENTED |
| HLR02 | OS(WINDOWS,LINUX) | IMPLEMENTED |
| HLR03 | NOTEPAD | IMPLEMENTED |
| HLR04 | MS WORD | IMPLEMENTED |
| HLR05 | 4 GB RAM | IMPLEMENTED |

**ARCHITECTURE:**

****

**IMPLEMENTATION**:

#include <stdio.h>  
#include<time.h>  
#include<stdlib.h>  
char board[9]={' ',' ',' ',' ',' ',' ',' ',' ',' '};  
void show\_board()  
{  
    printf("   |   |   \n");  
    printf(" %c | %c | %c\n",board[0],board[1],board[2]);  
    printf("   |   |   \n");  
    printf("-----------\n");  
    printf("   |   |   \n");  
    printf(" %c | %c | %c\n",board[3],board[4],board[5]);  
    printf("   |   |   \n");  
    printf("-----------\n");  
    printf("   |   |   \n");  
    printf(" %c | %c | %c\n",board[6],board[7],board[8]);  
    printf("   |   |   \n");  
}  
int countb(char a)  
{  
    int sum=0;  
    for(int i=0;i<9;i++)  
    {  
        if(board[i]==a)  
            sum=sum+1;  
    }  
    return sum;  
}  
void computerchoice()  
{  
    srand(time(0));  
    int ch;  
    do{  
        ch=rand()%10;  
    }while(board[ch]!=' ');  
    board[ch]='O';  
}  
void playerchoice()  
{  
    while(1)  
    {  
        int ch;  
        printf("Enter position \n");  
        scanf("%d",&ch);  
        ch--;  
        if(ch<0 || ch>8)  
        {  
            printf("Invalid position \n");  
        }  
        else if(board[ch]!=' ')  
        {  
            printf("Enter unoccupied position \n");  
        }  
        else{  
            board[ch]='X';  
            break;  
        }  
    }  
}  
void player2choice()  
{  
    while(1)  
    {  
        int ch;  
        printf("Enter position \n");  
        scanf("%d",&ch);  
        ch--;  
        if(ch<0 || ch>8)  
        {  
            printf("Invalid position \n");  
        }  
        else if(board[ch]!=' ')  
        {  
            printf("Enter unoccupied position \n");  
        }  
        else{  
            board[ch]='O';  
            break;  
        }  
    }  
}  
char winner()  
{  
    //checking winner horizontally  
    if(board[0]==board[1] && board[1]==board[2] && board[0]!=' ')  
    {  
        return board[0];  
    }  
    if(board[3]==board[4] && board[4]==board[5] && board[3]!=' ')  
    {  
        return board[3];  
    }  
    if(board[6]==board[7] && board[7]==board[8] && board[6]!=' ')  
    {  
        return board[6];  
    }  
    //checking vertically  
    if(board[0]==board[3] && board[3]==board[6] && board[0]!=' ')  
    {  
        return board[0];  
    }  
    if(board[1]==board[4] && board[4]==board[7] && board[1]!=' ')  
    {  
        return board[1];  
    }  
    if(board[2]==board[5] && board[5]==board[8] && board[2]!=' ')  
    {  
        return board[2];  
    }  
    //checking diagonally  
     if(board[0]==board[4] && board[4]==board[8] && board[0]!=' ')  
    {  
        return board[0];  
    }  
    if(board[2]==board[4] && board[4]==board[6] && board[1]!=' ')  
    {  
        return board[2];  
    }  
    if(countb('X')+countb('O')<9)  
    {  
        return 'C';  
    }  
    else{  
        return 'D';  
    }  
}  
void multiplayer(){  
    char p1[20],p2[20];  
    printf("Enter player1 name:");  
    scanf("%s",p1);  
    printf("Enter player2 name:");  
    scanf("%s",p2);  
    while(1){  
        system("cls");  
        show\_board();  
        if(countb('X')==countb('O'))  
        {  
            playerchoice();  
        }  
        else  
        {  
            player2choice();  
        }  
        char win=winner();  
        if(win=='X')  
        {  
            system("cls");  
            show\_board();  
            printf("%s won\n",p1);  
            break;  
        }  
        else if(win=='O')  
        {  
            system("cls");  
            show\_board();  
            printf("%s won\n",p2);  
            break;  
        }  
        else if(win=='D')  
        {  
            system("cls");  
            show\_board();  
            printf("match drawn\n");  
            break;  
        }  
    }  
}  
         
void compvsplayer()  
{  
    char name[20];  
    printf("Enter player name ");  
    scanf("%s",name);  
    while(1)  
    {  
        system("cls");  
        show\_board();  
        if(countb('X')==countb('O'))  
        {  
            playerchoice();  
        }  
        else  
        {  
            computerchoice();  
        }  
        char win=winner();  
        if(win=='X')  
        {  
            system("cls");  
            show\_board();  
            printf("%s won\n",name);  
            break;  
        }  
        else if(win=='O')  
        {  
            system("cls");  
            show\_board();  
            printf("Computer won\n");  
            break;  
        }  
        else if(win=='D')  
        {  
            system("cls");  
            show\_board();  
            printf("match drawn\n");  
            break;  
        }  
    }  
}  
  
int main()  
{  
    int ch=1;  
    while(ch){  
        system("cls");  
        printf("1.Player vs Player\n");  
        printf("2.Player vs computer\n");  
        printf("3.Exit\n");  
        int op;  
        scanf("%d",&op);  
        switch(op)  
        {  
            case 1:  
                multiplayer();  
                break;  
            case 2:  
                compvsplayer();  
                break;  
            case 3:  
                ch=0;  
                break;  
            default:  
                printf("Enter valid option\n");  
        }  
    }  
    return 0;  
}

**TESTPLAN AND OUTPUT:**

**sh: 1: cls: not found**

**1.Player vs Player**

**2.Player vs computer**

**3.Exit**

**1**

**Enter player1 name:john**

**Enter player2 name:dinesh**

**sh: 1: cls: not found**

**| |**

**| |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**Enter position**

**1**

**sh: 1: cls: not found**

**| |**

**X | |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**Enter position**

**3**

**sh: 1: cls: not found**

**| |**

**X | | O**

**| |**

**-----------**

**| |**

**| |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**Enter position**

**2**

**sh: 1: cls: not found**

**| |**

**X | X | O**

**| |**

**-----------**

**| |**

**| |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**Enter position**

**4**

**sh: 1: cls: not found**

**| |**

**X | X | O**

**| |**

**-----------**

**| |**

**O | |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**Enter position**

**5**

**sh: 1: cls: not found**

**| |**

**X | X | O**

**| |**

**-----------**

**| |**

**O | X |**

**| |**

**-----------**

**| |**

**| |**

**| |**

**Enter position**

**6**

**sh: 1: cls: not found**

**| |**

**X | X | O**

**| |**

**-----------**

**| |**

**O | X | O**

**| |**

**-----------**

**| |**

**| |**

**| |**

**Enter position**

**8**

**sh: 1: cls: not found**

**| |**

**X | X | O**

**| |**

**-----------**

**| |**

**O | X | O**

**| |**

**-----------**

**| |**

**| X |**

**| |**

**john won**

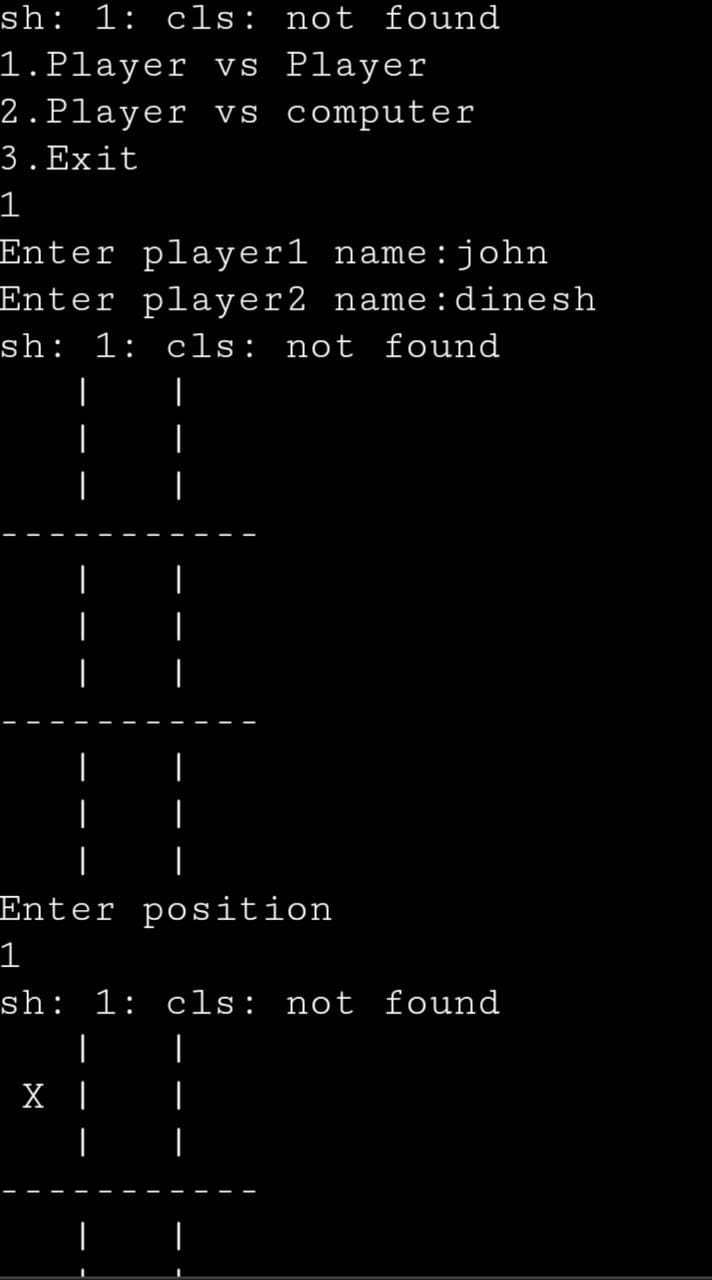
**sh: 1: cls: not found**

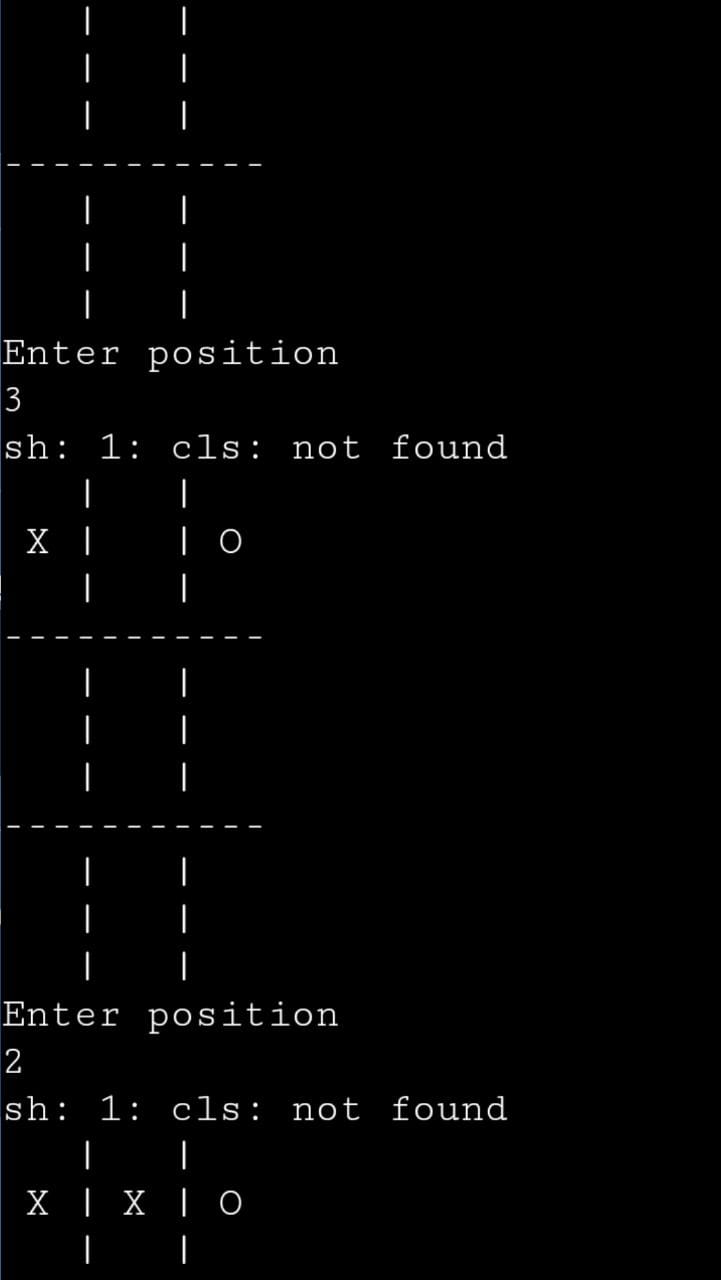
**1.Player vs Player**

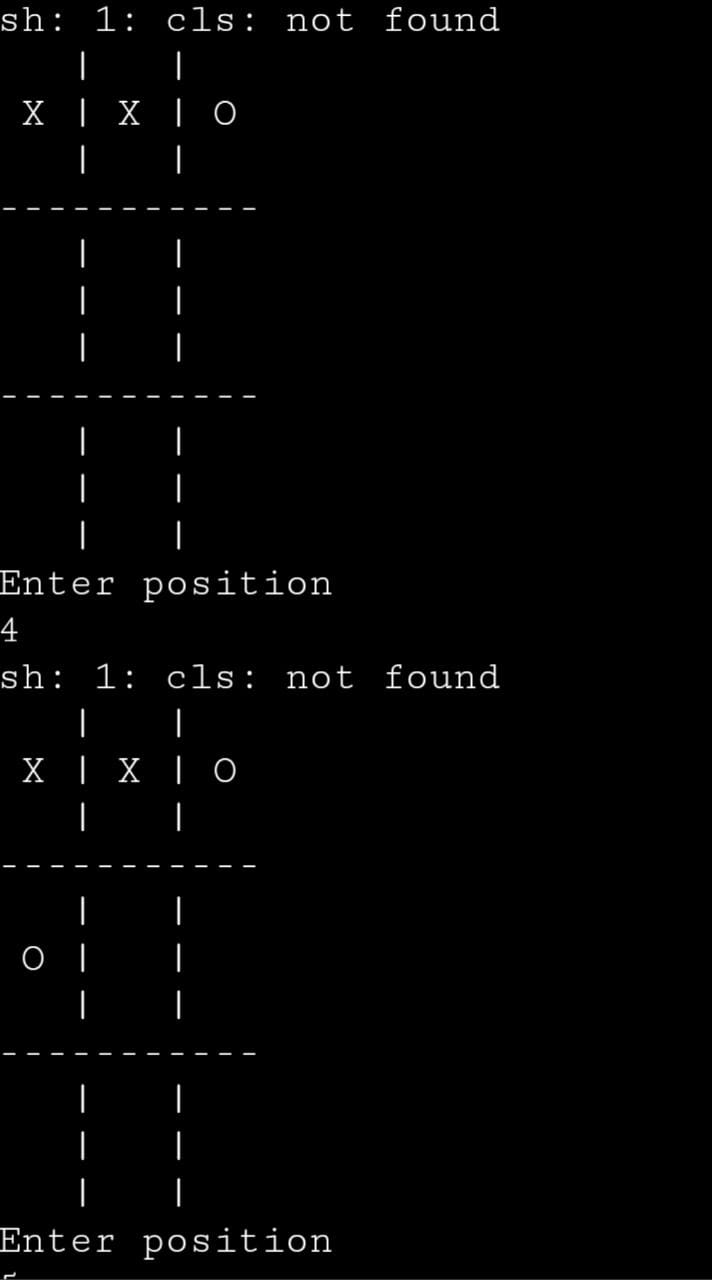
**2.Player vs computer**

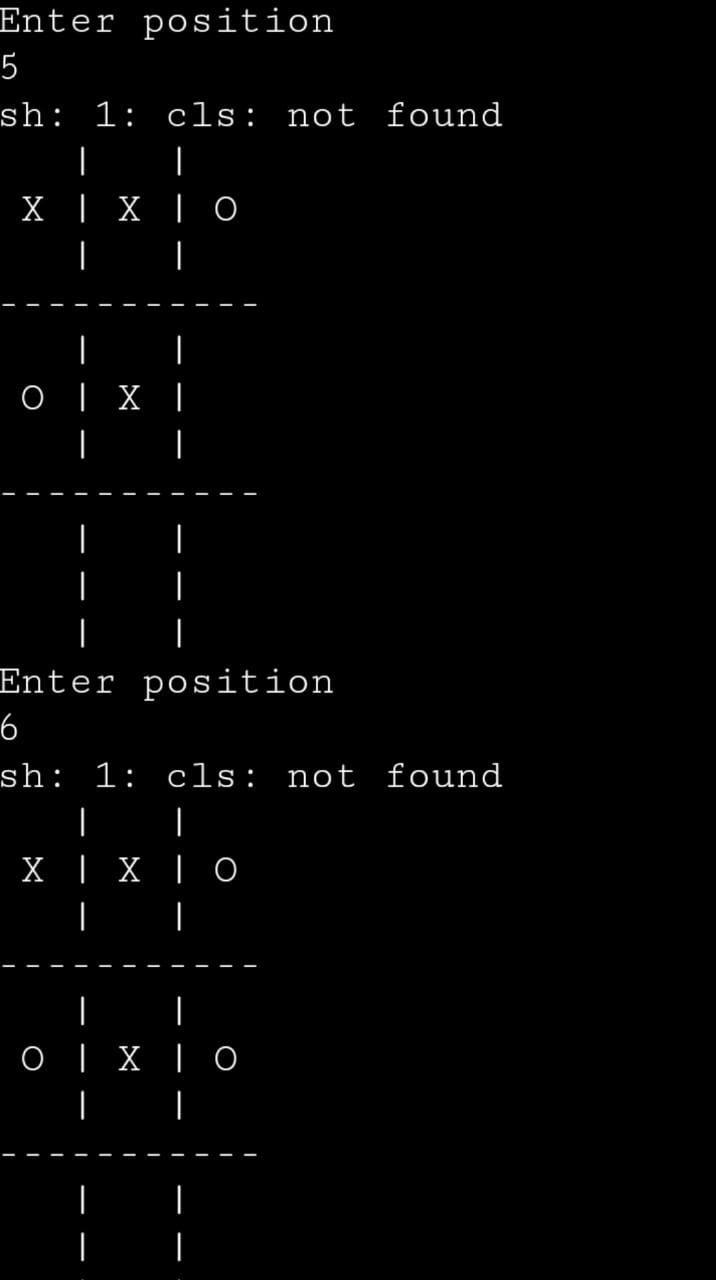
**3.Exit**

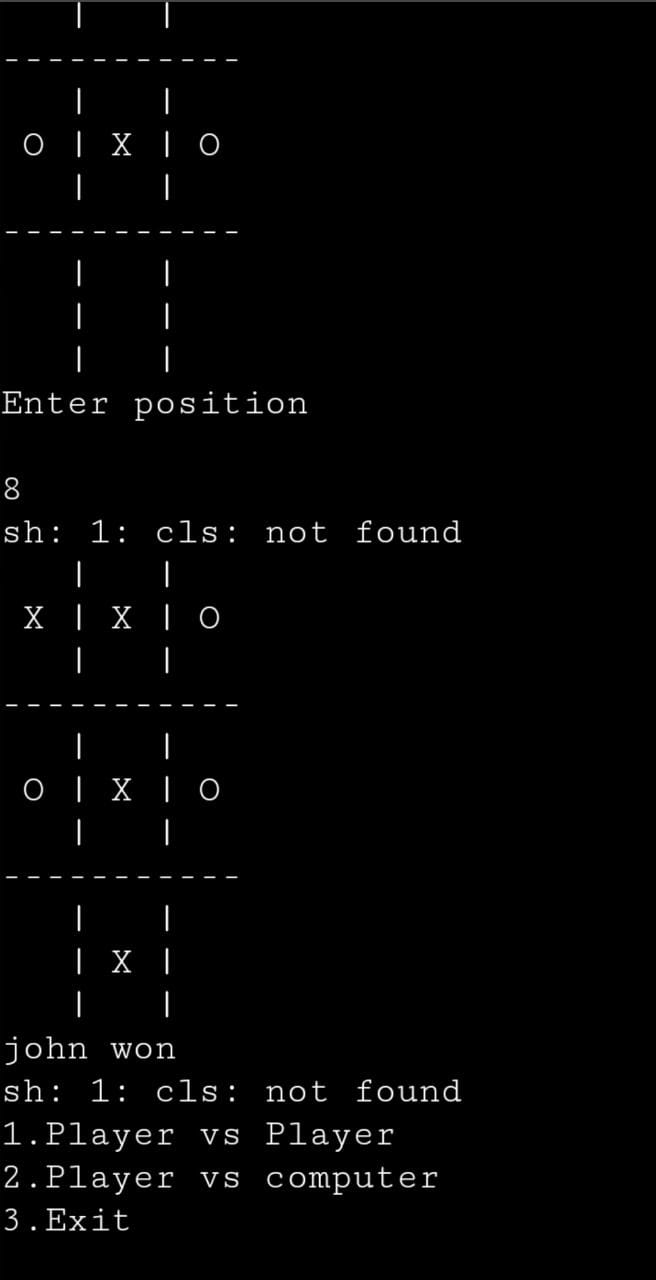
**OUTPUT:**

****

****

****

****

****