**Experiment No 2.4**

**Student Name: Ayushi Aggarwal UID: 21MCA2806**

**Branch: MCA Section/Group: 8B**

**Semester: 2nd Date of Performance: 04-04-2022**

**Subject Name: Computing Aptitude Subject Code: 21CAP-654**

1. **Aim/Overview of the practical:**

Create any interesting game using C++ code -like tic tac toe, snake and ladder, find a word, maze and shuffling cards etc.

1. **Code for experiment/practical:**

#include<bits/stdc++.h>

using namespace std;

#define COMPUTER 1

#define HUMAN 2

#define SIDE 3 // Length of the board

#define COMPUTERMOVE 'O'

#define HUMANMOVE 'X'

void showBoard(char board[][SIDE])

{

    printf("\n\n");

    printf("\t\t\t  %c | %c  | %c  \n", board[0][0],

                             board[0][1], board[0][2]);

    printf("\t\t\t--------------\n");

    printf("\t\t\t  %c | %c  | %c  \n", board[1][0],

                             board[1][1], board[1][2]);

    printf("\t\t\t--------------\n");

    printf("\t\t\t  %c | %c  | %c  \n\n", board[2][0],

                             board[2][1], board[2][2]);

    return;

}

void showInstructions()

{

    printf("\t\t\t  Tic-Tac-Toe\n\n");

    printf("Choose a cell numbered from 1 to 9 as below"

            " and play\n\n");

    printf("\t\t\t  1 | 2  | 3  \n");

    printf("\t\t\t--------------\n");

    printf("\t\t\t  4 | 5  | 6  \n");

    printf("\t\t\t--------------\n");

    printf("\t\t\t  7 | 8  | 9  \n\n");

    printf("-\t-\t-\t-\t-\t-\t-\t-\t-\t-\n\n");

    return;

}

void initialise(char board[][SIDE], int moves[])

{

    srand(time(NULL));

    for (int i=0; i<SIDE; i++)

    {

        for (int j=0; j<SIDE; j++)

            board[i][j] = ' ';

    }

    for (int i=0; i<SIDE\*SIDE; i++)

        moves[i] = i;

    random\_shuffle(moves, moves + SIDE\*SIDE);

    return;

}

void declareWinner(int whoseTurn)

{

    if (whoseTurn == COMPUTER)

        printf("COMPUTER has won\n");

    else

        printf("HUMAN has won\n");

    return;

}

bool rowCrossed(char board[][SIDE])

{

    for (int i=0; i<SIDE; i++)

    {

        if (board[i][0] == board[i][1] &&

            board[i][1] == board[i][2] &&

            board[i][0] != ' ')

            return (true);

    }

    return(false);

}

bool columnCrossed(char board[][SIDE])

{

    for (int i=0; i<SIDE; i++)

    {

        if (board[0][i] == board[1][i] &&

            board[1][i] == board[2][i] &&

            board[0][i] != ' ')

            return (true);

    }

    return(false);

}

bool diagonalCrossed(char board[][SIDE])

{

    if (board[0][0] == board[1][1] &&

        board[1][1] == board[2][2] &&

        board[0][0] != ' ')

        return(true);

    if (board[0][2] == board[1][1] &&

        board[1][1] == board[2][0] &&

         board[0][2] != ' ')

        return(true);

    return(false);

}

bool gameOver(char board[][SIDE])

{

    return(rowCrossed(board) || columnCrossed(board)

            || diagonalCrossed(board) );

}

void playTicTacToe(int whoseTurn)

{

    char board[SIDE][SIDE];

    int moves[SIDE\*SIDE];

    initialise(board, moves);

    showInstructions();

    int moveIndex = 0, x, y;

    while (gameOver(board) == false &&

            moveIndex != SIDE\*SIDE)

    {

        if (whoseTurn == COMPUTER)

        {

            x = moves[moveIndex] / SIDE;

            y = moves[moveIndex] % SIDE;

            board[x][y] = COMPUTERMOVE;

            printf("COMPUTER has put a %c in cell %d\n",

                    COMPUTERMOVE, moves[moveIndex]+1);

            showBoard(board);

            moveIndex ++;

            whoseTurn = HUMAN;

        }

        else if (whoseTurn == HUMAN)

        {

            x = moves[moveIndex] / SIDE;

            y = moves[moveIndex] % SIDE;

            board[x][y] = HUMANMOVE;

            printf ("HUMAN has put a %c in cell %d\n",

                    HUMANMOVE, moves[moveIndex]+1);

            showBoard(board);

            moveIndex ++;

            whoseTurn = COMPUTER;

        }

    }

    if (gameOver(board) == false &&

            moveIndex == SIDE \* SIDE)

        printf("It's a draw\n");

    else

    {

        if (whoseTurn == COMPUTER)

            whoseTurn = HUMAN;

        else if (whoseTurn == HUMAN)

            whoseTurn = COMPUTER;

        declareWinner(whoseTurn);

    }

    return;

}

int main()

{

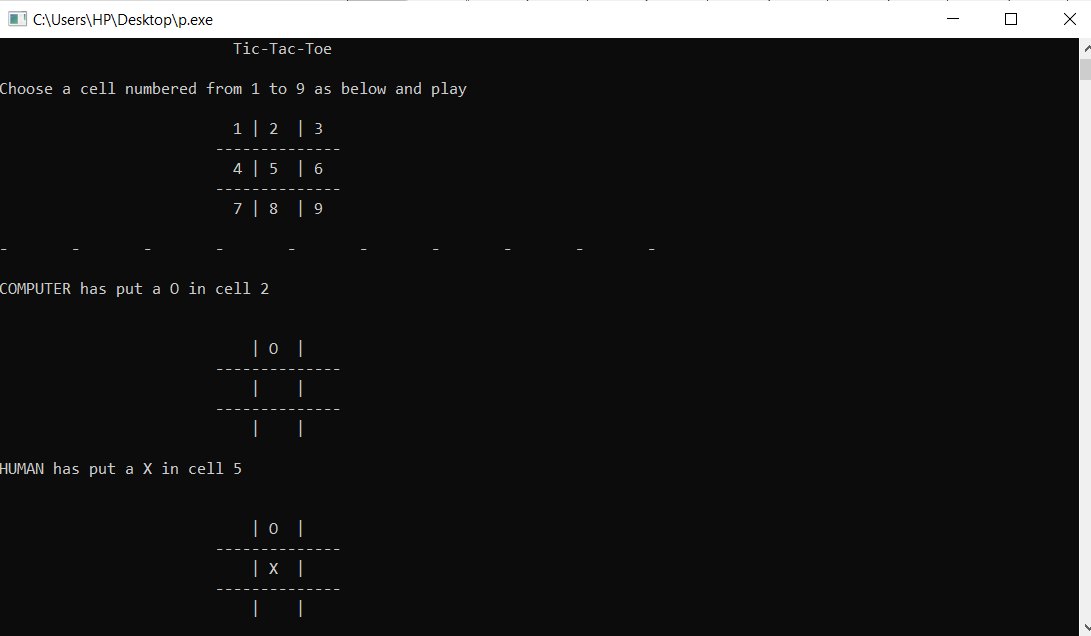
    playTicTacToe(COMPUTER);

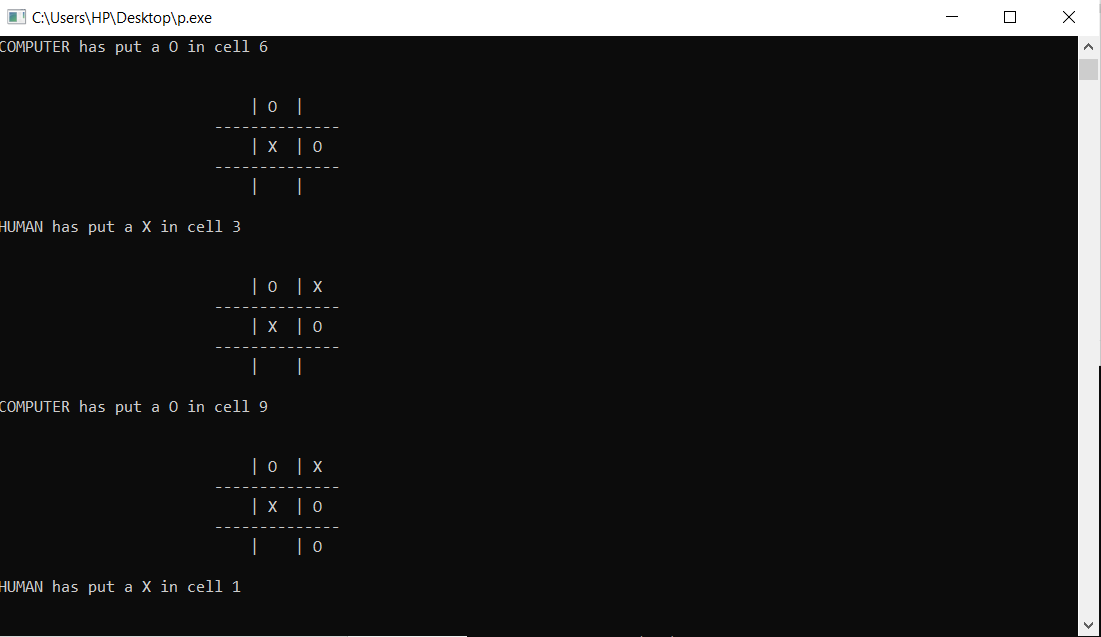
    return (0);

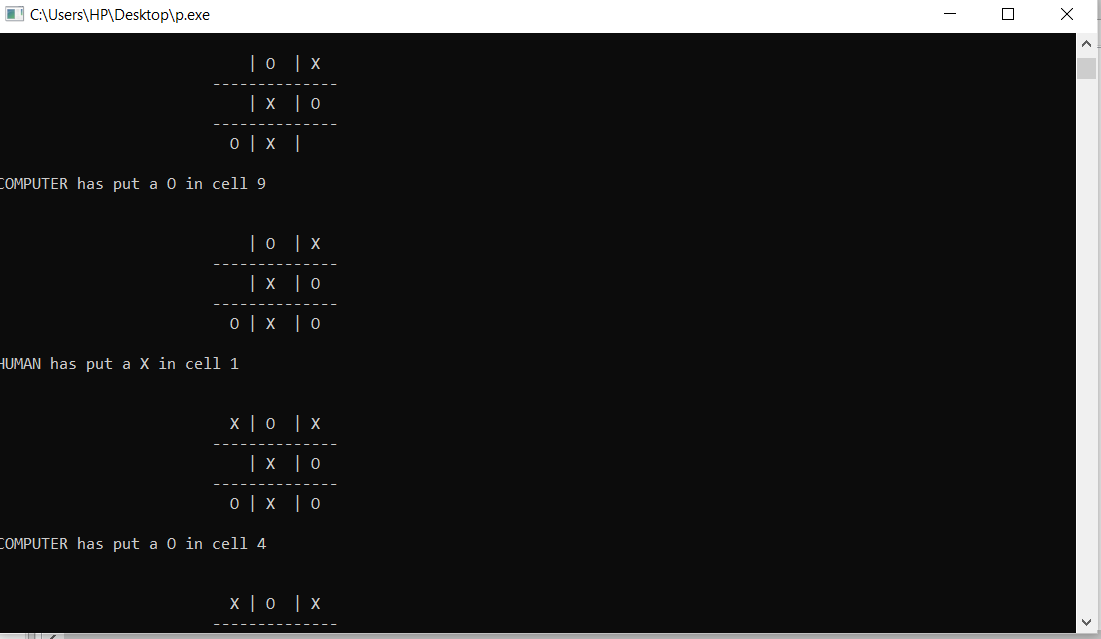
}

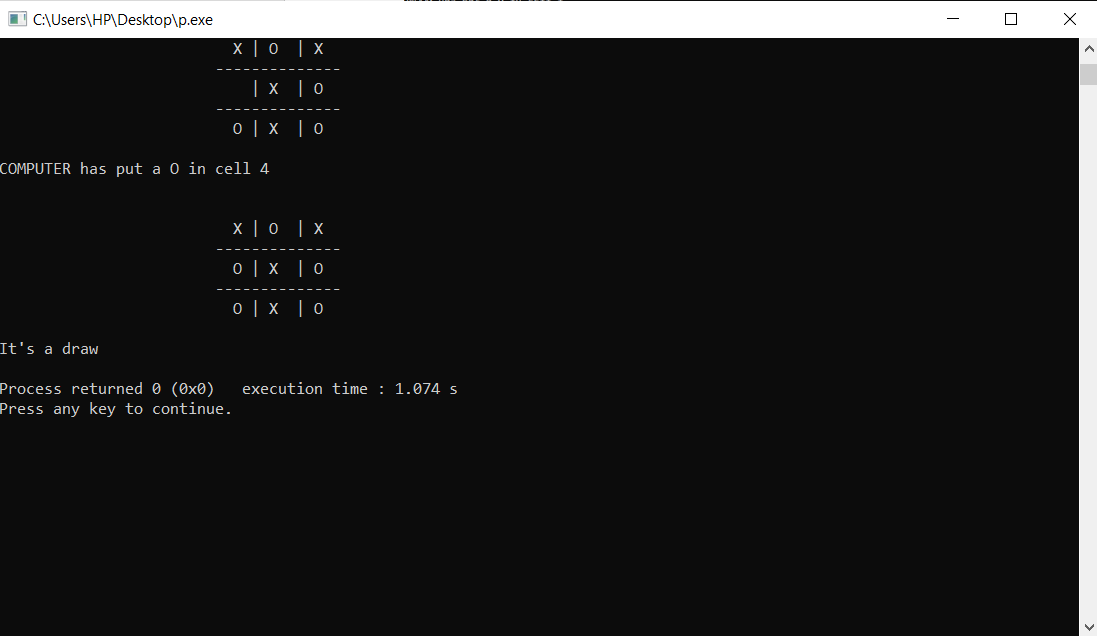
1. **Result/Output/Writing Summary:**

**Output: -**









**Evaluation Grid:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Demonstration and Performance |  | 5 |
| 2. | Worksheet |  | 10 |
| 3. | Post Lab Quiz |  | 5 |