DATA STRUCTURE AND ALGORITHM LAB

TIC TAC TOE GAME

Project Report By

Muhammad Umair.

CS-191050

3B

Group Member:

Muhammad Umair CS-191050

Hasan Ali Waqas CS-191037

Syed Owais Ali CS-191002

Table of Contents

1. Introduction :

The first print reference to "**noughts and crosses**," the British name for the game, appeared in 1864. The first print reference to a game called "tick-tack-toe" occurred in 1884 but referred to a children's game played on a slate. **Tic-tac-toe** have **noughts and crosses** , or **Xs and Os** is a game for two players, *X* and *O*, who take turns marking the spaces in a 3×3 grid. **Tic-tac-toe** game is a paper-pencil game for two players, *X* and *O*, who take turns marking the spaces in a 3×3 grid. Now we can made tic tac toe on console without using paper pencil. We help the players without making the 3×3 grid on paper now players just mark the options.

2. Scope :

In this project we use the one concept of DSA (Data Structure And Algorithm) which is linked list.

Linked List :

In linked list further we use singly linked list.

Singly Linked List :

In this project I use singly linked list.

Algorithms Used In This Project :

Following are the algorithms used in this project are :

1. Insert Data.

2. Update Data.

3. Print Board.

4. Check Winner.

Language Used In This Project :

In this project I use c++ language.

Platform Used In This Project :

In this project I use Visual Studio Code as platform.

How To Win Tic Tac Toe Game :

Tic Tac Toe, also known as *"Noughts and Crosses"* or *"X's and O's"*, is a solved game. This means there is a known, mathematically proven strategy to follow for the best result each game. In Tic Tac Toe, two players who follow the right strategy will always tie, with neither player winning. Against an opponent who doesn't know this strategy, however, you can still win whenever they make a mistake. Once your friends pick up on your strategy, try a more difficult version of the rules.

If you don't know how to play tic tac toe, learn the basic rules.

Play your first X in a corner :

Most experienced tic tac toe players put the first "X" in a corner when they get to play first. This gives the opponent the most opportunities to make a mistake. If your opponent responds by putting an O *anywhere* besides the center, you can guarantee a win.

Try to win if your opponent plays the first O in the center :

If your opponent plays their first O in the center, you have to wait for them to make a mistake before you can win. If they continue to play correctly, they can guarantee a tie. Here are your two options for your second move, followed by instructions on how to win if they make certain moves (if they don't, just keep blocking their plays and the game will be a tie):

- Place your second X in the opposite corner from your first, so there's a line going "X O X" diagonally across the board. If they respond with an O in one of the other corners, you can win! Place your third X in the last empty corner, and your opponent won't be able to block you from winning with your fourth X.
- **Or**, place your second X on an edge square (not a corner), *not* touching your first X. If your opponent puts down an O in the corner that's *not* next to your X, you

can use your third X to block their move and automatically win with your fourth X.

Win automatically if your opponent plays his first O in any square besides the center :

If your opponent puts his first O in any square *besides* the center, you can win. Respond by putting your second X in any other corner, with an empty space in between the two X's.

Place your third X so you have two possible winning moves :

Most of the time, your opponent will see that you have two X's in a row and block you. (If not, just win by making a row of three X's.) After this happens, there should be an empty square that is in line with both your first and your second X, with no enemy O's blocking that line. Put your third X in this square.

Win with your fourth X :

After your third X, there are two empty squares that will win you the game if an X goes into one of them. Since your opponent can only make one move, he can only block one of those squares. Write your fourth X into the square he didn't block, and you've won the game!

Force a draw if the opponent starts in the corner :

If the opponent plays first and starts with an O in a corner, always put your first X in the center. Your second X should be placed on an edge, *not* a corner, *unless* you need to block your opponent from getting three in a row. Using this strategy, every game should be a draw. Theoretically, you can win from this position, but your opponent would have to make a huge mistake, such as not seeing that you have two X's in a row.

- In this section, your opponent is still playing O's, but remember they get to play first this time.

Force a draw when the opponent starts in the center:

When your opponent starts by putting down an O in the center, place your first X in a corner. After that, just keep blocking your opponent from scoring and the game will be a draw.There is essentially no way for you to win from this position, unless your opponent stops trying to win or stop you from winning!

Try to win if the opponent starts at the edge:

Most of the time, your opponent will start with one of the moves above. However, if your opponent puts down the first O on an edge, not on a corner or center, you have a small chance to win. Put your first X in the center. If your opponent puts the second O on the opposite edge, making a row or column that reads O-X-O, put your second X in a corner. Then, if your opponent puts the third O in the edge that is adjacent to your X, making a line that reads O-X-O, put your third X in the empty square to block their row of two O's. From here, you can always win with your fourth X.

- If at any point, your opponent doesn't make the exact move described above, you'll have to settle for a draw. Just start blocking their moves and neither of you will win.

Try these out if your tic tac toe games always end in draws :

It might be fun for a while to be unbeatable at tic tac toe, but even without this article your friends might figure out how to stop you from winning. Once that happens, every single game of tic tac toe you play with them will be a draw — ugh. But you can still use basic tic tac toe rules to play games that aren't as easily solved. Try them out below.

Play mental tic tac toe :

The rules are exactly the same as tic tac toe, but there's no board! Instead, each player says their moves aloud, and pictures the board in their head. You can still use all the strategy advice in this article, but it can be difficult to concentrate on that when you're trying to remember where the X's and O's are.

- Agree on a system for describing moves. For instance the first word is the row (top, middle, or bottom) and the second word is the column (left, middle, or right).

5. Screen Shots (attach screenshot of source code and output) :

CODE

node.h

```h
#ifndef NODE_H
#define NODE_H

struct Node
{
    Node* next;
    int data;

    Node(int data);

};
#endif

```

node.cpp

```cpp
#include <iostream>
#include "node.h"

Node::Node(int data)
{
    this->data = data;      // is mein hum data ko data ke equal karen ge.
    this->next = NULL;      // is mein hum next ko null ke equal karen ge.
}

```

linklist.h

```h
#ifndef LINKLIST_H
#define LINKLIST_H
#include <iostream>
#include "node.h"

class LinkList
{
    private:
        Node* head;
        Node* current;

    public:
        LinkList();
        void insertData(int data);
        void updateData(int enterValue, char playerTurn);
        void printBoard();
        bool checkWinner(std::string player1 , std::string player2);
        void rematchAndNewgame();
};
#endif

```

linklist.cpp

```cpp
#include <iostream>
#include "linklist.h"

LinkList::LinkList()    // linked list ke constructor ke andr.
{
    this->head = NULL;       // head ko bhi hum null ke equal karden ge.
    this->current = NULL;     // aur current ko bhi null ke equal karden ge.
}

void LinkList::insertData(int data)         //inserting node at end
{
    Node* temp= new Node(data);
    if (head == NULL)             //agr linked list ke andr koi bhi node nai hai tu ye first node create hogi.
    {
        head =temp;            // aur is mein jo new node hogi for example first node wo head hojae gi.
    }
    else                          // or agr phele se first node hai tu wo direct else mein jae ga.
    {
        Node* current = head;            // sub se phele hum head ko current karden ge.
        while (current->next != NULL)     // then while ka loop us waqt tak chale ga jab tak current->next ko null na mil j
        {
            current = current->next;      // aur loop ke andr jab tak current ko null nai mile ga tab tak wo loop ke andr c
        }
        current->next = temp;         // aur jese hi current->next ko null mil jae ga wo loop ke bahir ajae ga then null k
    }
}

```

linklist.cpp

```
void LinkList::updateData(int enterValue , char playerTurn)    // is function mein board ka data update hoga jese value ke
{
    Node* current = head;             // sab se phele hum current ko head ke equal karden ge .

    while(current->data != enterValue)      // then while ke loop ke andr current->data jab tak entered value ke equal nai
    {
        current = current->next;      // is loop ke andr jab tak current data entered value ke equal nai hoga tab tak curre
    }
    current->data = playerTurn;       // aur jese hi current data enterd value ke equal hojae ga tu wo loop ke bahir ajae g
}

void LinkList::printBoard()          // is print ke function ke andr hum board print kar rae hain.
{
    system("cls");
    std::cout << "\t\t\t\t----------------------------------------" << std::endl;
    std::cout << "\t\t\t\t|          TIC TAC TOE GAME            |" << std::endl;       // title of game.
    std::cout << "\t\t\t\t----------------------------------------" << std::endl;
    std::cout << "\t\t\t\t\tPlayer 1 (X)  -  Player 2 (O)" << std::endl << std::endl;    // players info.
    std::cout << std::endl;

    Node* current = head;         // sab se phele hum head ko current karden ge

    while(current != NULL)
    {
        std::cout << "\t\t\t\t";
        for(int i=0; i<3; i++)          // ye for ka loop first row ke liye hai.
        {
            if(current->data == 79) // if ascii value is 79
            {
                std::cout << "\t" << "O";   // agr second player ki bari hai tu wo O print karde ga aur phir direct label p
                goto label;       // agr if ki condition true hoti hai tu direct label pe jump hojae ga.
            }
```

linklist.cpp

```
            else if(current->data == 88)  // if  ascii value is 88
            {
                std::cout << "\t" << "X";  // agr first player ki bari hai tu wo X print karde ga aur phir direct label pe
                goto label;            // agr else-if ki condition true hoti hai tu direct label pe jump hojae ga.
            }

            std::cout << "\t" << current->data;

            label:                        // aur jab label pe jump ho ga tu next condition chale gi.
            current = current->next;      // idr hum next node mein jane ke liye current ko current->next ke equal karen g

            if(i<2)
            {
                std::cout << "   |";
            }
        }

        std::cout << "\t\t\t\t";
        std::cout << std::endl;
        std::cout << "\t\t\t\t      --------------------";
        std::cout << std::endl;
        std::cout << "\t\t\t\t";

        for(int i=0; i<3; i++)              // ye for ka loop second row ke liye hai.
        {
            if(current->data == 79) // if ascii value is 79
            {
                std::cout << "\t" << "O";   // agr second player ki bari hai tu wo O print karde ga aur phir direct label p
                goto label2;              // agr if ki condition true hoti hai tu direct label2 pe jump hojae ga.
            }

            else if(current->data == 88)  // if 88
```

linklist.cpp

```
            std::cout << "\t" << current->data;

            label2:                     // aur jab label2 pe jump ho ga tu next condition chale gi.
            current = current->next;       // idr hum next node mein jane ke liye current ko current->next ke equal karen g
            if(i<2)
            {
                std::cout << "   |";
            }
        }

        std::cout << "\t\t\t\t";
        std::cout << std::endl;
        std::cout << "\t\t\t\t      --------------------";
        std::cout << std::endl;
        std::cout << "\t\t\t\t";

        for(int i=0; i<3; i++)                // ye for ka loop third row ke liye hai.
        {
            if(current->data == 79) // if ascii value is 79
            {
                std::cout << "\t" << "O";        // agr second player ki bari hai tu wo O print karde ga aur phir direct la
                goto label3;                  // agr if ki condition true hoti hai tu direct label3 pe jump hojae ga.
            }

            else if(current->data == 88)  // if 88
            {
                std::cout << "\t" << "X";        // agr first player ki bari hai tu wo X print karde ga aur phir direct lab
                goto label3;                    // agr else-if ki condition true hoti hai tu direct label3 pe jump hojae g
            }

            std::cout << "\t" << current->data;
```

linklist.cpp

```
}

bool LinkList::checkWinner(std::string player1, std::string player2)        // is check ke function mein hum board check ka
{
    Node* current = head;     // sab se phele hum head ko current karen ge.

    // Sab se phele hum first row check karen ge.

    std::cout << std::endl << std::endl;
    if(current->data == current->next->data && current->data == current->next->next->data)    // sab se current jo head ke
    {
        if(current->data == 'X')        // agr current->data 'X' ke equal hoga tu player 1 winner hoga.
        {
            std::cout << "\n\t\t\t\t\t\t" << player1<< " [X] WIN!!!" << std::endl;
        }
        else                        // agr current->data 'O' ke equal hoga tu player 2 winner hoga.
        {
            std::cout << "\n\t\t\t\t\t\t" << player2<< " [O] WIN!!!" << std::endl;
        }
        return true;
    }

    //checking first column.

    else if(current->data == current->next->next->next->data && current->data == current->next->next->next->next->next->nex
    {
        if(current->data == 'X')    // agr current->data 'X' ke equal hoga tu player 1 winner hoga.
        {
            std::cout << "\n\t\t\t\t\t\t" << player1<< " [X] WIN!!!" << std::endl;
        }
        else                        // agr current->data 'O' ke equal hoga tu player 2 winner hoga.
        {
            std::cout << "\n\t\t\t\t\t\t" << player2<< " [O] WIN!!!" << std::endl;
```

linklist.cpp

```cpp
    }

    //Then hum draw check karen ge.

    current = head;       // sab se phele hum current ko head ke equal karden ge.

    bool goFlag = false;    // then hum bool ke variable ko false ke equal karen ge.

    while(current != NULL)        // while ka loop us waqt tak chale ga  jab tak use null na mil jae.the while ki condition
    {
        if(current->data == 'X' || current->data == 'O')     // age current->data equal hota hai hai X ya O ke tu wo true r
        {
            goFlag = true;
        }
        else             // agr current->data X ya O ke equal nai hota tu wo else ke andr jae ga aur goflag mein false retu
        {
            goFlag = false;
        }
        if(goFlag == false)    // agr go flag false ke equal hota tu to draw wali condition break hojae gi
        {
            break;
        }
        current = current->next;   // aur agr us ko current->data pe X ya O mil jata hai tu wo current ko current->next ke
    }
    if(goFlag == true)   // sare boxes check karne ke baad agr goflag true hota hai ti wo print karde ga match draw warna r
    {
        std::cout << "\n\t\t\t\t\t  ------GAME DRAW------" << std::endl;
        return true;
    }
    return false;
}

```

main.cpp

```cpp
#include<iostream>
#include "linklist.h"
#include "node.h"
#include <cstdlib>
#include <ctime>

int main()
{
    LinkList ll;                // yahan hum link list ka object create kar rae hain.
    char playerTurn = ' ';
    int enterValue= { };
    char temp = ' ';
    std::string player1;
    std::string player2;
    int choice = 0;

    ll.insertData(1);
    ll.insertData(2);
    ll.insertData(3);           // ye hum values insert kar rae hain.
    ll.insertData(4);
    ll.insertData(5);
    ll.insertData(6);
    ll.insertData(7);
    ll.insertData(8);
    ll.insertData(9);

    do              // sab se phele hum do while ka loop create karen ge jis mein sab se phele program ik dafa run hoga the
    {
        system("cls");
        std::cout<<"\t\t\t---------------------------------------"<<std::endl;
        std::cout<<"\t\t\t|          TIC TAC TOE GAME           |"<<std::endl;
        std::cout<<"\t\t\t---------------------------------------"<<std::endl;
```

main.cpp

main.cpp > main()

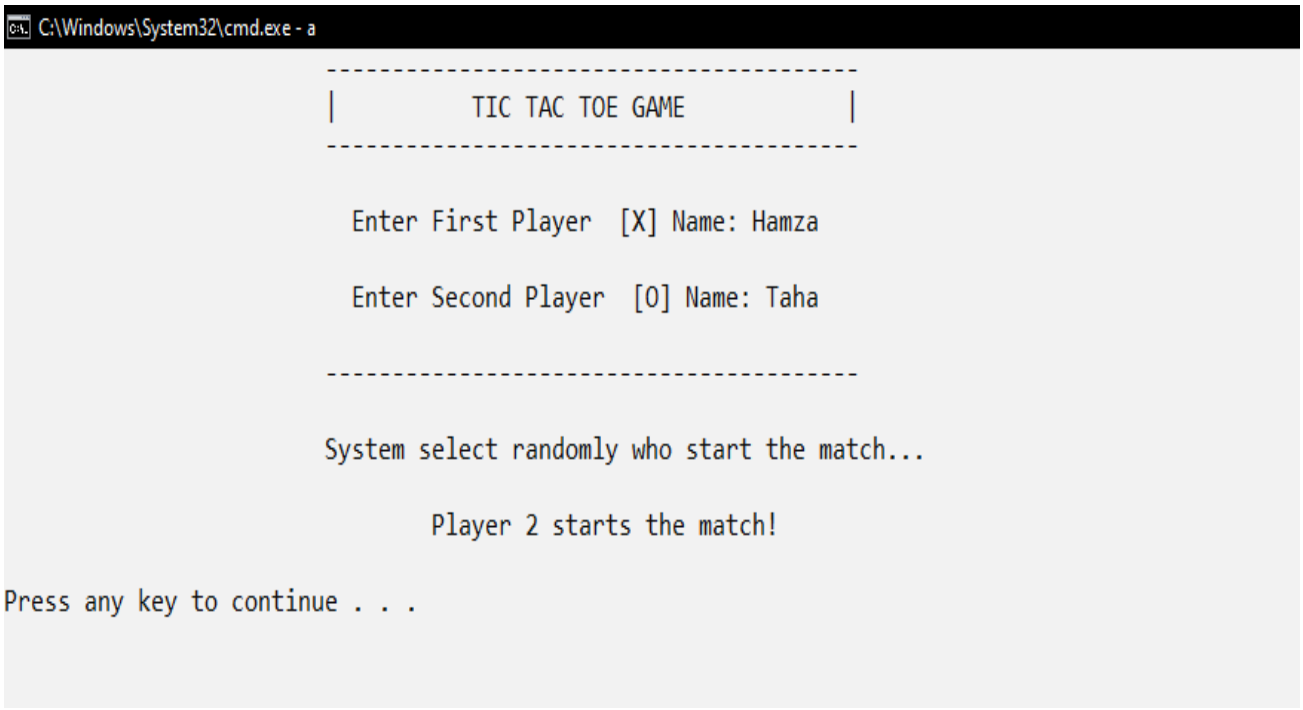
```
58
59              system("pause");
60              system("cls");
61
62              bool flag = false;   // yahan hum sab se phele flag ke variable ko false ke equal karen ge.
63              while(true)       // aur while ke loop ke andr true hoga tu wo while ke loop ke andr ajae ga
64              {
65                  if(flag == true)    // aur while ke loop ke andr ane ke baad wo if ke check ke andr jae ga agr if ki condit
66                  {
67                      break;          // aur if ke check ke andr ane ke baad condition break hojae gi.
68                  }
69
70                  ll.printBoard();       // aur phir linked list ke andr print ka function call hoga jo board print kare ga.
71                  std::cout<<std::endl<<std::endl;
72
73                  if(playerTurn == 'X')        // is if ke check ki condition check hogi agr jese is se phele bari 'X' ki thi
74                  {
75                      std::cout<<"\n\t\t\t\t   "<<player1<<"'s [X] turn: ";
76                      temp = 'O';             // aur jese hi if ki condition true hogi tu temp ke andr 'O' chala jae ga.
77                  }
78                  else     // is else ki condition check hogi agr jese is se phele bari 'O' ki thi aur turn ke variable ke an
79                  {
80                      std::cout<<"\n\t\t\t\t   "<<player2<<"'s [O] turn: ";
81                      temp = 'X';             // aur jese hi else ki condition true hogi tu temp ke andr 'X' chala jae ga.
82                  }
83                  std::cin>>enterValue;
84
85                  while(enterValue < 1 || enterValue > 10 )   //checking players choice is in the range of '1' to '10'
86                  {
87                      std::cout<<"invalid Choice.....Please Enter Again"<<std::endl;
88                      std::cin>>enterValue;
89                  }
90
```

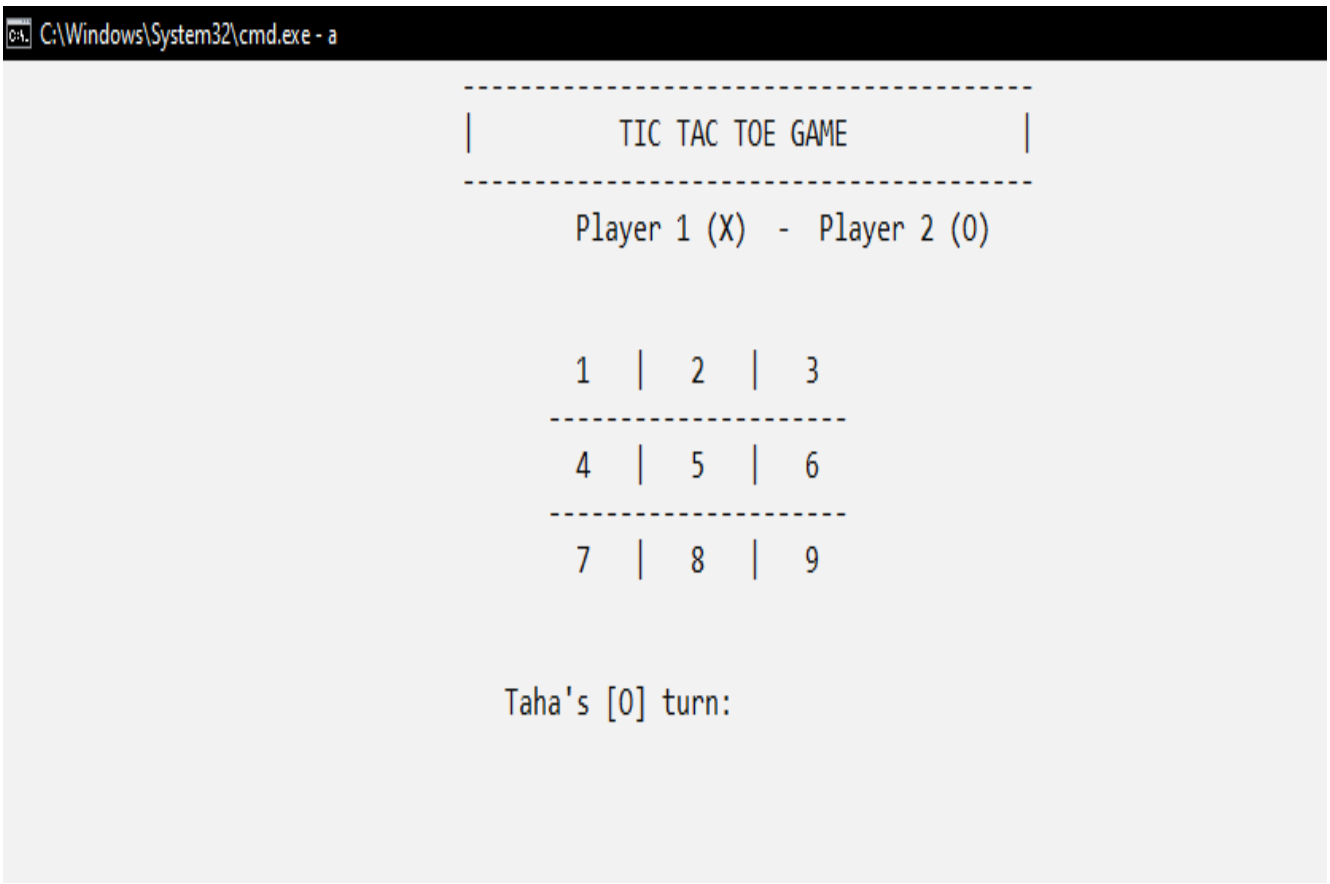
main.cpp

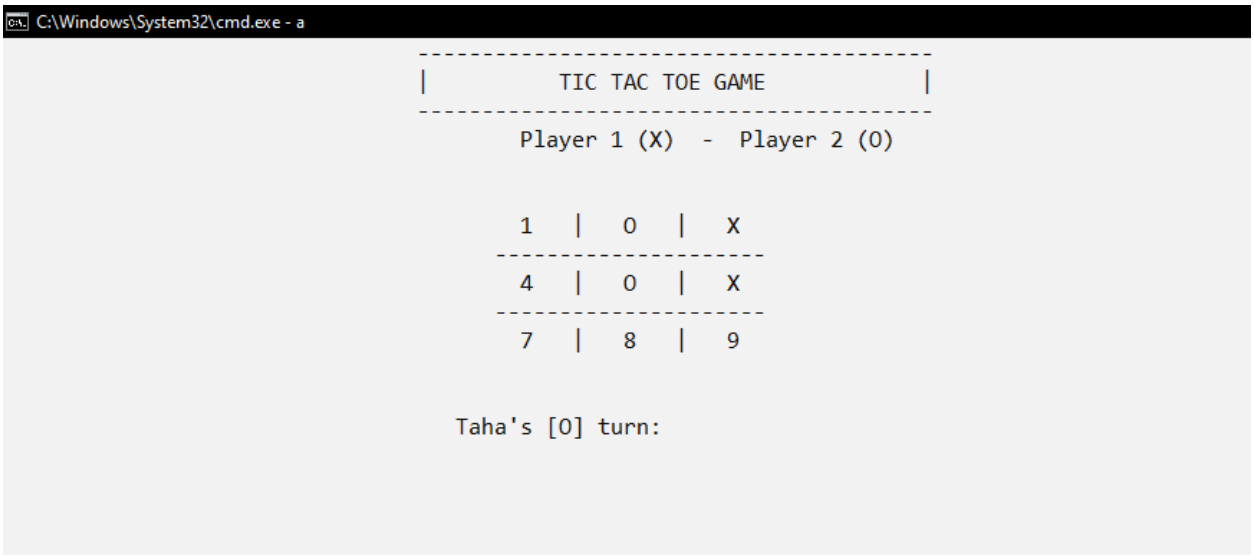
main.cpp > main()

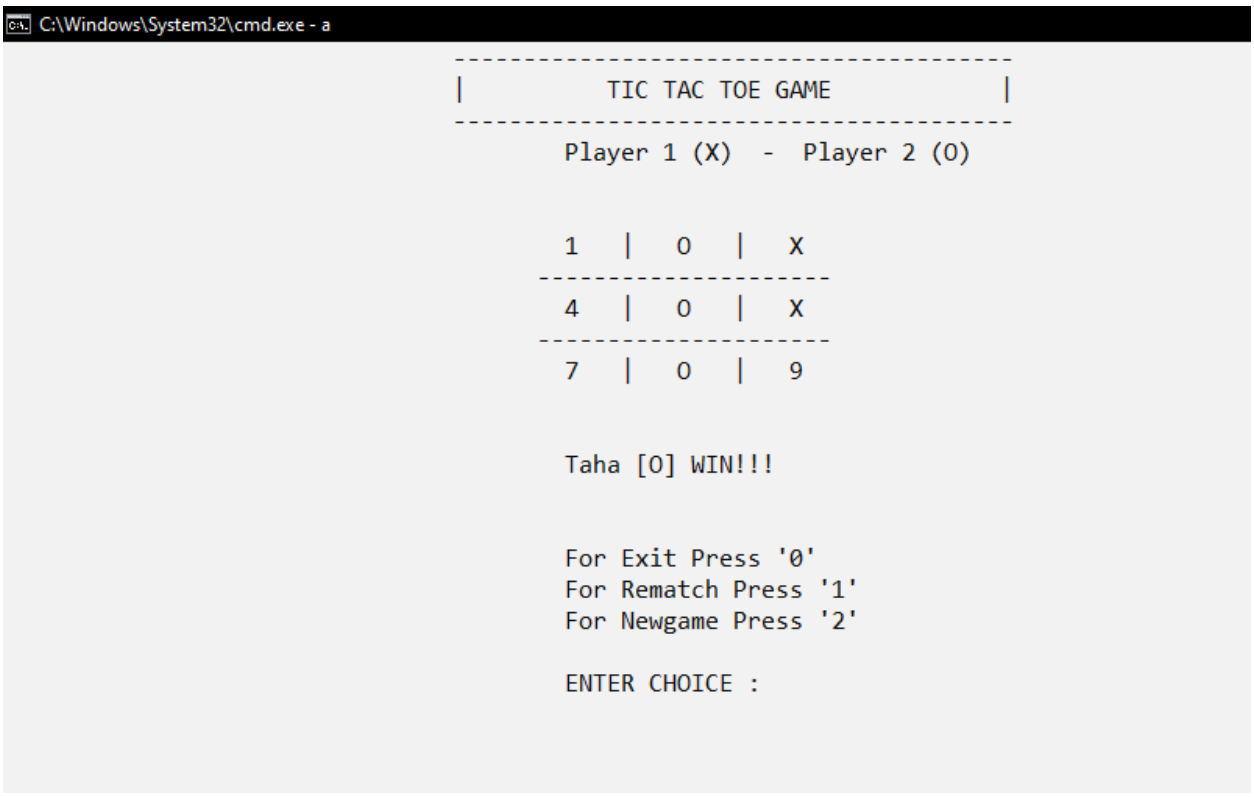
```
26
27      do              // sab se phele hum do while ka loop create karen ge jis mein sab se phele program ik dafa run hoga the
28      {
29          system("cls");
30          std::cout<<"\t\t\t---------------------------------------"<<std::endl;
31          std::cout<<"\t\t\t|          TIC TAC TOE GAME           |"<<std::endl;
32          std::cout<<"\t\t\t---------------------------------------"<<std::endl;
33          std::cout<<std::endl;
34          std::cout<<"\t\t\t  Enter First Player  [X] Name: ";         // yahan player 1 ka name enter hoga.
35          std::cin>>player1;
36          std::cout<<"\n\t\t\t  Enter Second Player  [O] Name: ";         // yahan player 2 ka name enter hoga.
37          std::cin>>player2;
38          std::cout<<"\n\t\t\t---------------------------------------";
39
40          do           // second do while ke loop mein hum check karen ge kn sa player phele start kare ga aur ye ik dafa run
41          {
42              std::cout << "\n\n\t\t\tSystem select randomly who start the match...\n ";
43              srand(time(nullptr));
44              int dice = rand();    // yahan dice random numbers generate kare ga.
45
46              if(dice <= 3)    // age dice less than 3 ya equal to 3 hota hai tu player 1 ki bari hogi.
47              {
48                  int first = 1;
49                  std::cout << "\n\t\t\t\tPlayer 1 starts the match!\n";
50                  playerTurn = 'X';      // aur jese hi player 1 ki bari hogi turn ke variable ke andr 'X' chala jae ga.
51              }
52              else
53              {
54                  int first = 2;
55                  std::cout<<"\n\t\t\t\tPlayer 2 starts the match!\n\n";     // aur agr dice ke andr greater than 3 ata hai t
56                  playerTurn = 'O';      // aur jese hi player 2 start kare ga tu turn ke variable ke andr 'O' chala jae ga.
57              }
```

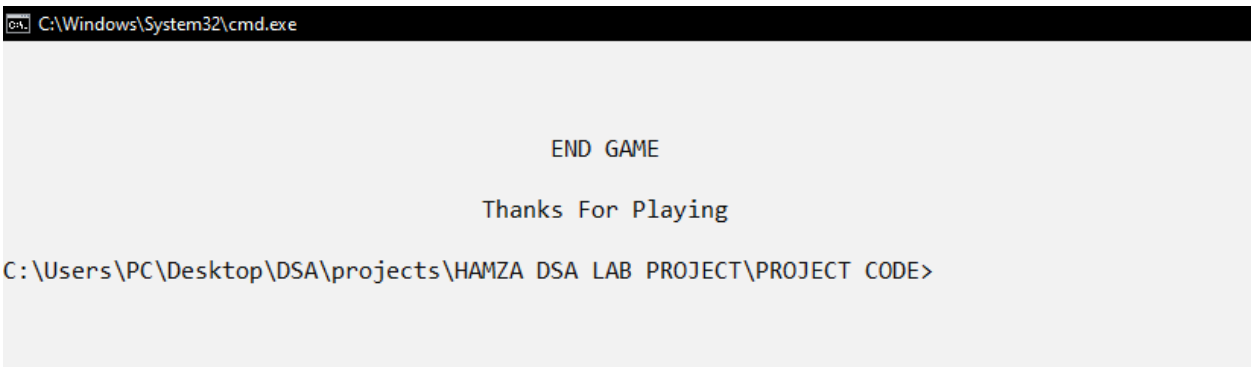
DEMO


C:\Windows\System32\cmd.exe - a
TIC TAC TOE GAME
Enter First Player [X] Name: Hamza
Enter Second Player [O] Name: Taha
System select randomly who start the match...
Player 2 starts the match!
Press any key to continue . . .


C:\Windows\System32\cmd.exe - a
TIC TAC TOE GAME
Player 1 (X) - Player 2 (O)
1 | 2 | 3
4 | 5 | 6
7 | 8 | 9
Taha's [O] turn:


C:\Windows\System32\cmd.exe - a
TIC TAC TOE GAME
Player 1 (X) - Player 2 (O)
1 | O | X
4 | O | X
7 | 8 | 9
Taha's [O] turn:


C:\Windows\System32\cmd.exe - a
TIC TAC TOE GAME
Player 1 (X) - Player 2 (O)
1 | O | X
4 | O | X
7 | O | 9
Taha [O] WIN!!!
For Exit Press '0'
For Rematch Press '1'
For Newgame Press '2'
ENTER CHOICE :


C:\Windows\System32\cmd.exe
END GAME
Thanks For Playing
C:\Users\PC\Desktop\DSA\projects\HAMZA DSA LAB PROJECT\PROJECT CODE>