ARTIFICIAL INTELLIGENCE

(AI)

BACHELOR's OF TECHNOLOGY

(Computer Science Engineering)

SUB MITTED BY:-

Pariva

Roll no:-1803546

Semester: - 7th

CTIEMT

SUBMITTED TO:-

Mr Abhishek Bhardwaj

(**Assistant professor)**

CSE/IT dept.

Table of Contents

20.				
21.				
22.				

Practical: 1

AIM: - Write a program for Tower of Hanoi.

```
#include<iostream>
#include<conio.h>
void towerHanoi(int n,char mid,char end,char beg)
{
   if(n==1){
      std::cout<<" move disc from"<<beg<<"------"<<end<<'\n';
   }
   else
   {
      towerHanoi(n-1,beg,end,mid);
      std::cout<<"move disc from"<<beg<<"-------"<<end<<'\n';
      towerHanoi(n-1,mid,beg,end);
   }
}
int main()
{

   int n;
   std::cout<<"enter the number of disc you want to display:\n";
   std::cin>>n;
   towerHanoi(n,'A','B','C');
   getch();
}
```

Output: - When n=5

```
enter the number of disc you want to display:
5
 move disc fromC-------B
move disc fromA--------B
 move disc fromB-------A
move disc fromC--------B
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B
move disc fromA--------B
 move disc fromB-------A
move disc fromC--------A
 move disc fromA-------C
move disc fromB--------A
 move disc fromC-------B
move disc fromA--------B
 move disc fromB-------A
move disc fromC--------B
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B
move disc fromA--------C
 move disc fromB-------A
move disc fromC--------A
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B
move disc fromA--------B
 move disc fromB-------A
move disc fromC--------B
```

```
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B
move disc fromA--------C
 move disc fromB------A
move disc fromC--------A
 move disc fromA------C
move disc fromB--------C
 move disc fromC-------B
move disc fromA--------B
 move disc fromB-------A
move disc fromC--------B
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B


...Program finished with exit code 0
Press ENTER to exit console.
```

When n=4

```
enter the number of disc you want to display:
4
 move disc fromA-------B
move disc fromC--------B
 move disc fromB-------C
move disc fromA--------B
 move disc fromC------A
move disc fromB--------A
 move disc fromA-------B
move disc fromC--------B
 move disc fromB-------C
move disc fromA--------C
 move disc fromC------A
move disc fromB--------C
 move disc fromA------B
move disc fromC--------B
 move disc fromB------C


...Program finished with exit code 0
Press ENTER to exit console.
```

When n=3

```
enter the number of disc you want to display:
3
 move disc fromC-------B
move disc fromA--------B
 move disc fromB-------A
move disc fromC--------B
 move disc fromA-------C
move disc fromB--------C
 move disc fromC-------B


...Program finished with exit code 0
Press ENTER to exit console.
```

When n=2

```
enter the number of disc you want to displa
2
 move disc fromA-------B
move disc fromC--------B
 move disc fromB-------C


...Program finished with exit code 0
Press ENTER to exit console.
```

Practical: 2

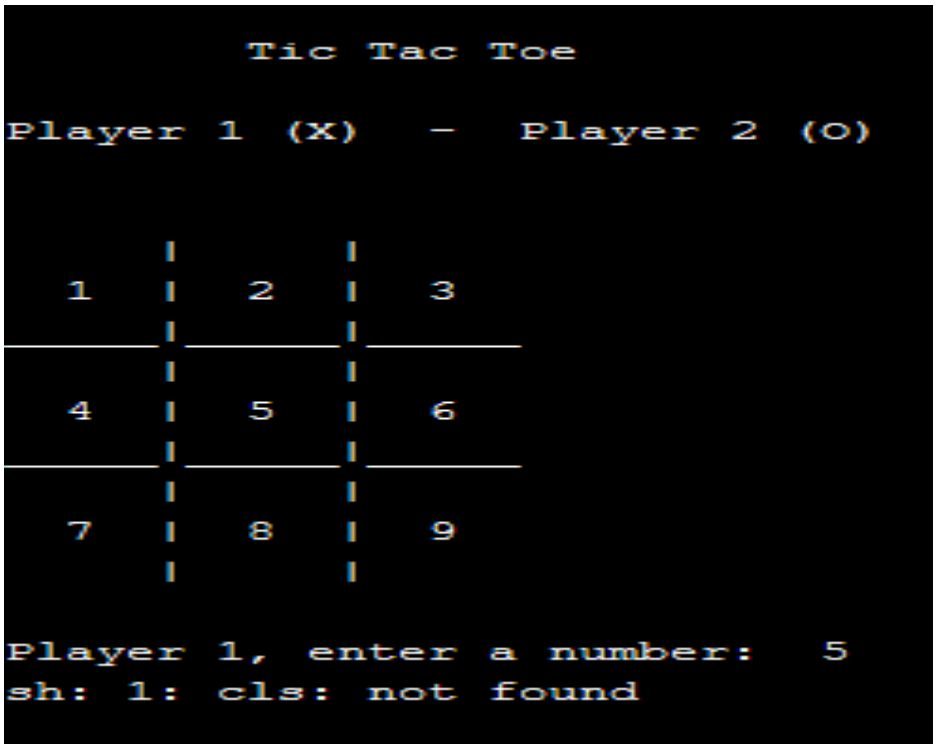
AIM: - Write a program of TIC TAC TOC in C++.

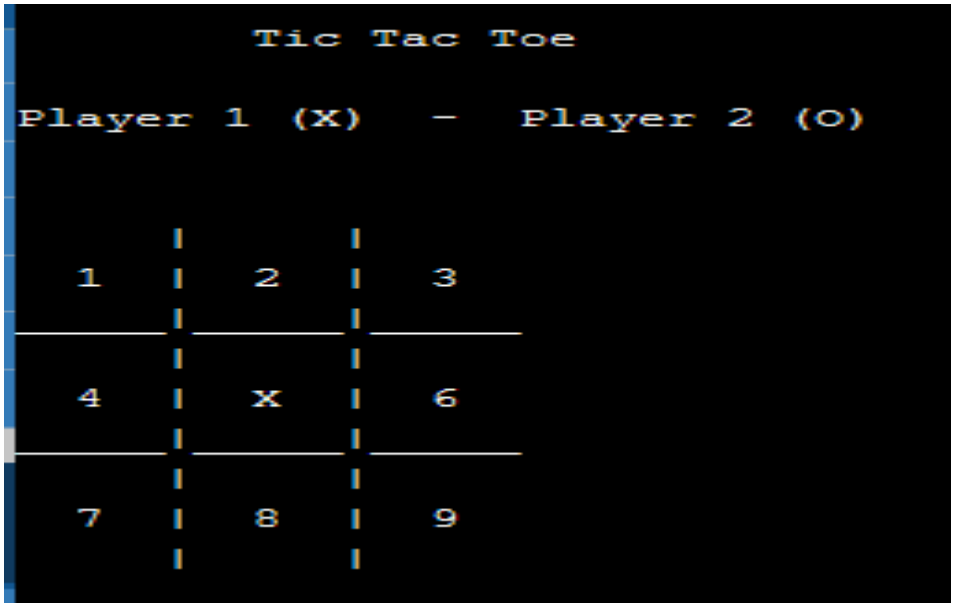
```
#include <iostream>
using namespace std;
char square[10] = {'o','1','2','3','4','5','6','7','8','9'};
int checkwin();
void board();
int main()
{
        int player = 1,i,choice;

   char mark;
   do
   {
      board();
      player=(player%2)?1:2;
      cout << "Player " << player << ", enter a number:  ";
      cin >> choice;
      mark=(player == 1) ? 'X' : 'O';
      if (choice == 1 && square[1] == '1')
         square[1] = mark;
      else if (choice == 2 && square[2] == '2')
         square[2] = mark;
      else if (choice == 3 && square[3] == '3')
         square[3] = mark;
      else if (choice == 4 && square[4] == '4')
square[4] = mark;
      else if (choice == 5 && square[5] == '5')
         square[5] = mark;
      else if (choice == 6 && square[6] == '6')
         square[6] = mark;
      else if (choice == 7 && square[7] == '7')
         square[7] = mark;
      else if (choice == 8 && square[8] == '8')
         square[8] = mark;
      else if (choice == 9 && square[9] == '9')
         square[9] = mark;
      else
      {
         cout<<"Invalid move ";

         player--;
         cin.ignore();
         cin.get();
      }
      i=checkwin();
```

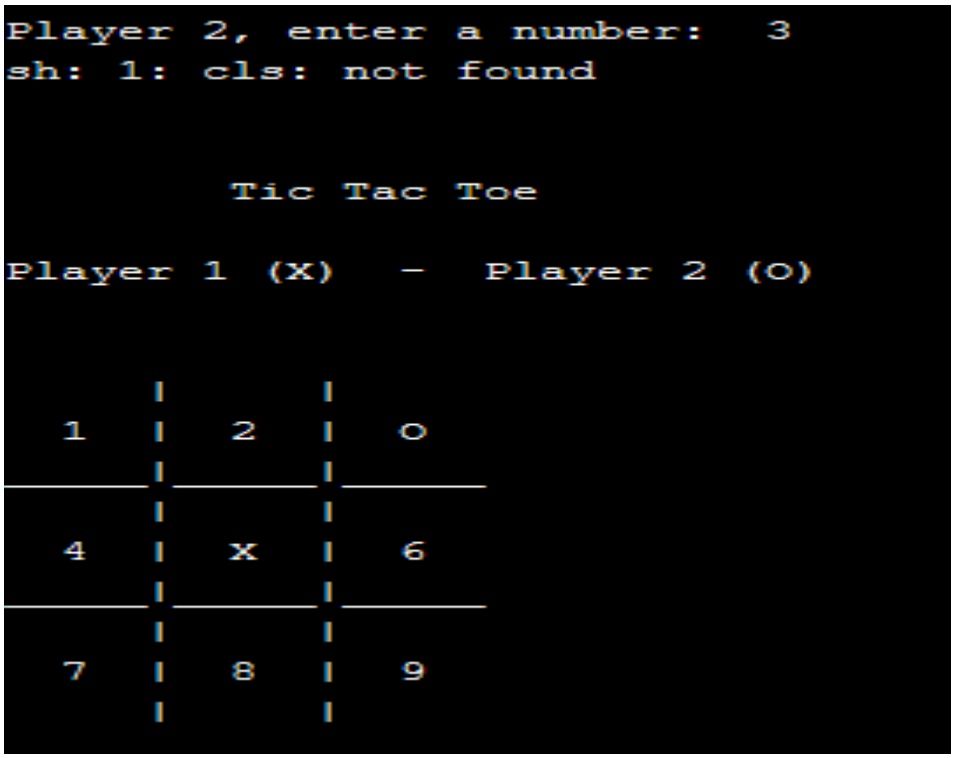
```
        player++;
    }while(i==-1);
    board();
    if(i==1)
        cout<<"==>\aPlayer "<<--player<<" win ";
    else
        cout<<"==>\aGame draw";
    cin.ignore();
    cin.get();
    return 0;
}

int checkwin()
{
    if (square[1] == square[2] && square[2] == square[3])

        return 1;
    else if (square[4] == square[5] && square[5] == square[6])

        return 1;
    else if (square[7] == square[8] && square[8] == square[9])

        return 1;
    else if (square[1] == square[4] && square[4] == square[7])

        return 1;
    else if (square[2] == square[5] && square[5] == square[8])

        return 1;
    else if (square[3] == square[6] && square[6] == square[9])

        return 1;
    else if (square[1] == square[5] && square[5] == square[9])

        return 1;
    else if (square[3] == square[5] && square[5] == square[7])

        return 1;
    else if (square[1] != '1' && square[2] != '2' && square[3] != '3'
                && square[4] != '4' && square[5] != '5' && square[6] != '6'
               && square[7] != '7' && square[8] != '8' && square[9] != '9')

        return 0;
    else
        return -1;
}
void board()
{
    system("cls");
    cout << "\n\n\tTic Tac Toe\n\n";
```

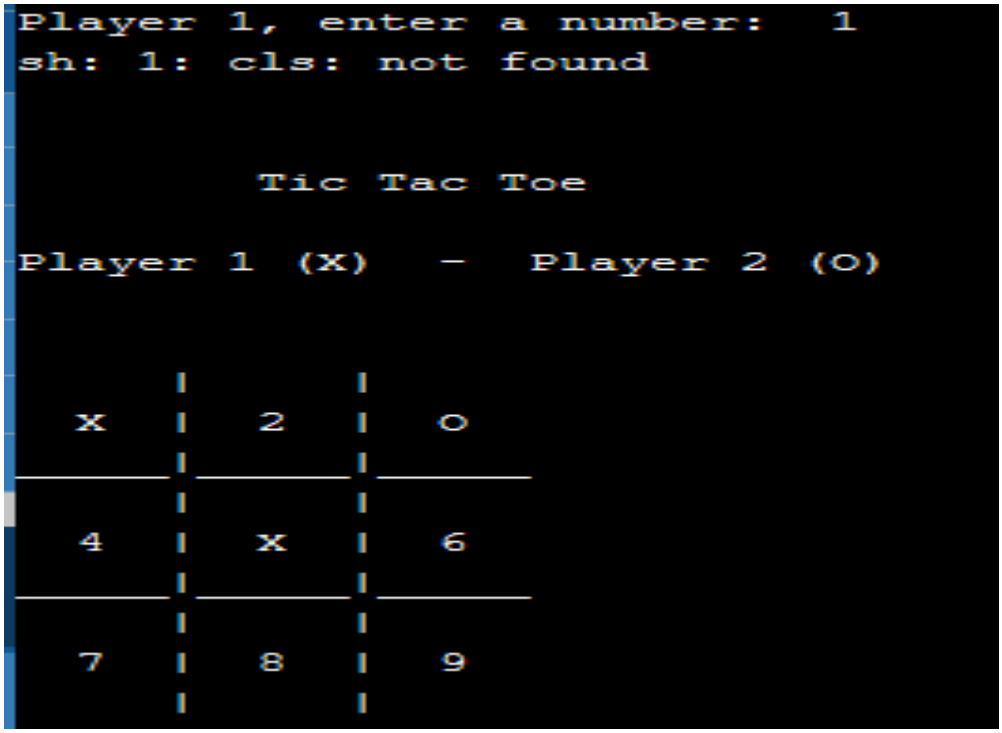
```
    cout << "Player 1 (X)  -  Player 2 (O)" << endl << endl;
    cout << endl;
    cout << "     |     |     " << endl;
    cout << "  " << square[1] << "  |  " << square[2] << "  |  " << square[3] << endl;
    cout << "_____|_____|_____" << endl;
    cout << "     |     |     " << endl;
    cout << "  " << square[4] << "  |  " << square[5] << "  |  " << square[6] << endl;
    cout << "_____|_____|_____" << endl;
    cout << "     |     |     " << endl;
    cout << "  " << square[7] << "  |  " << square[8] << "  |  " << square[9] << endl;
    cout << "     |     |     " << endl << endl;
}
```

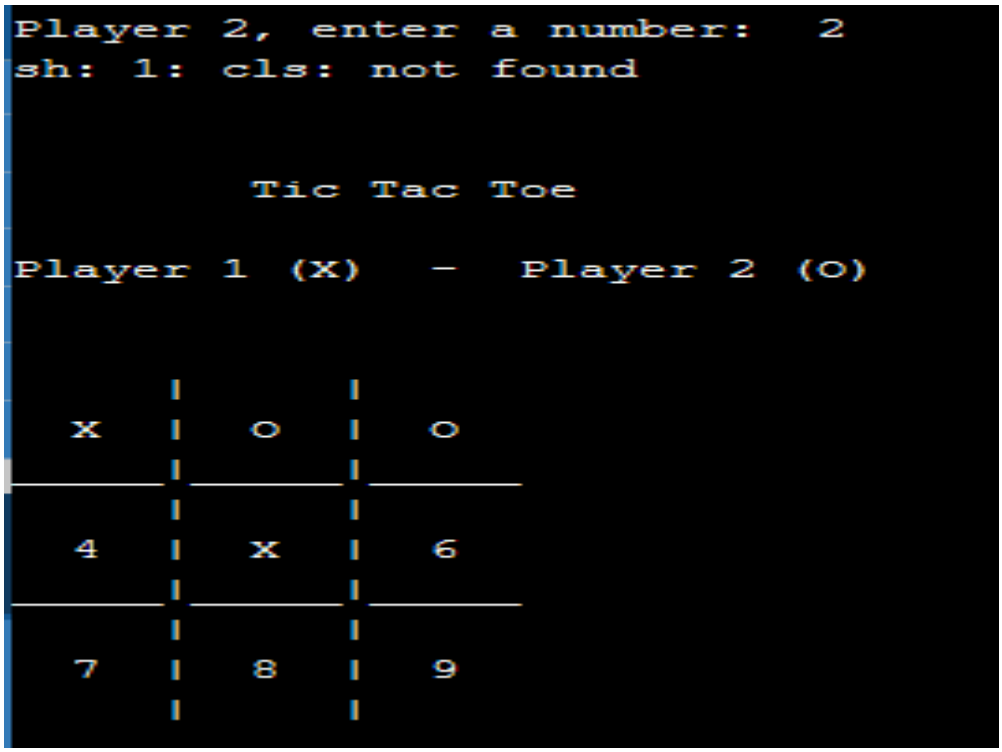
Output:-

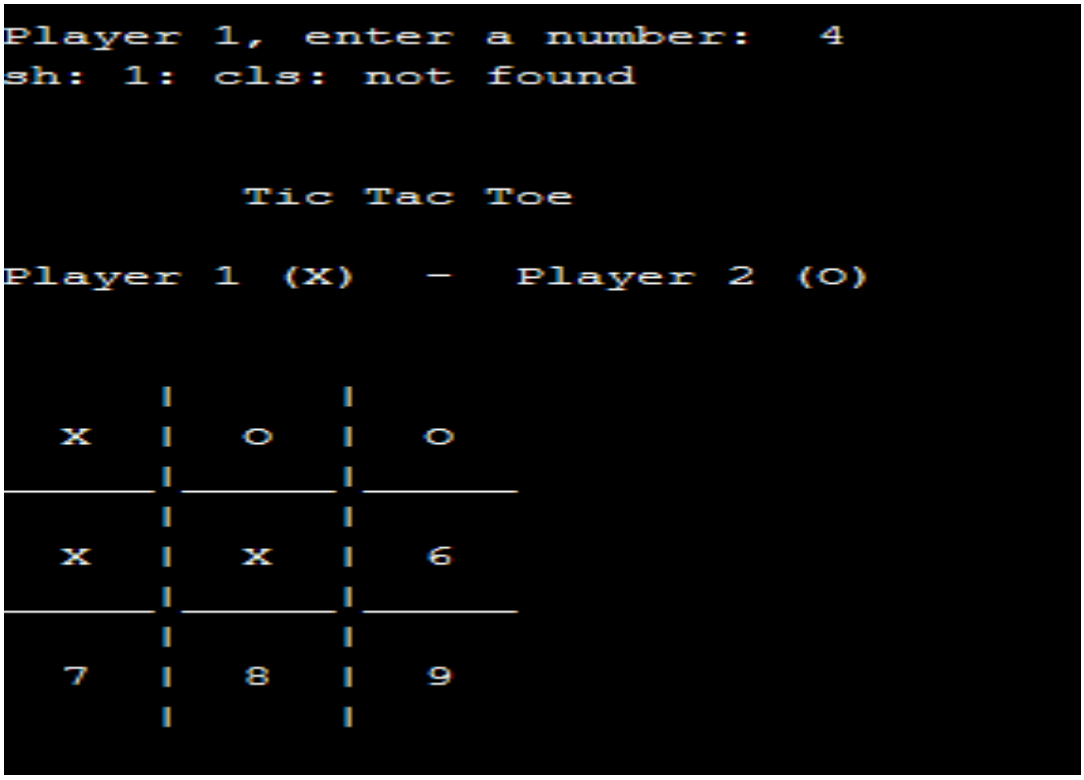


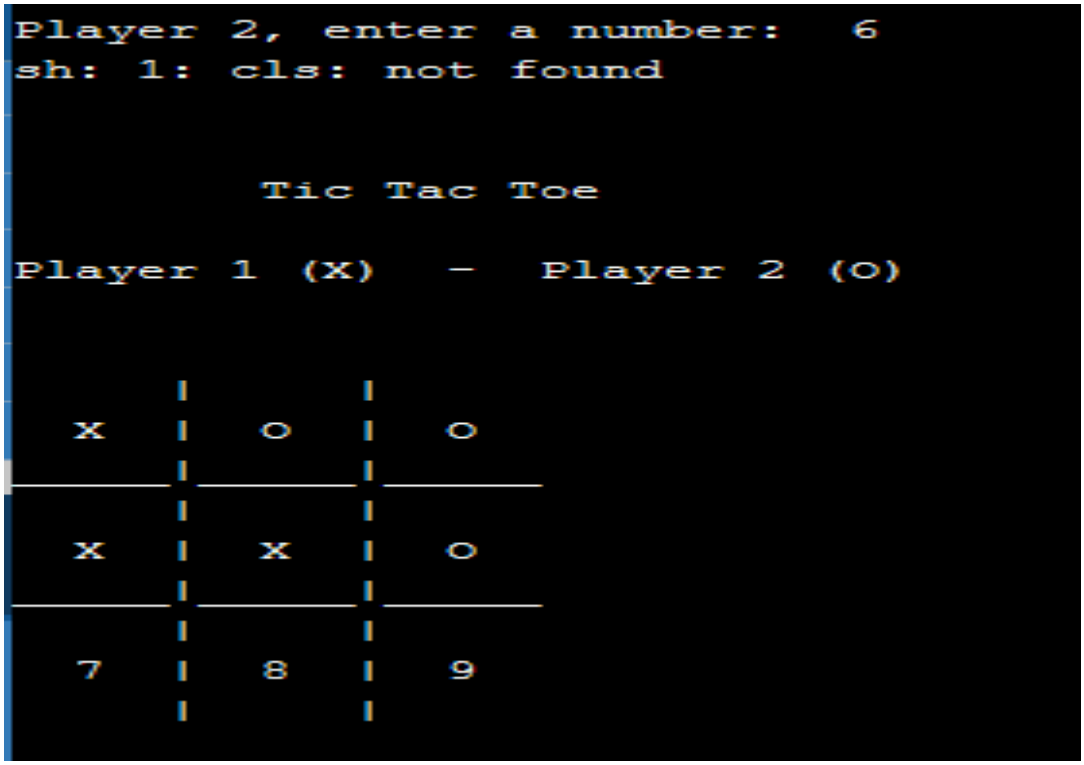

Tic Tac Toe
Player 1 (X) - Player 2 (O)
1 | 2 | 3
4 | X | 6
7 | 8 | 9

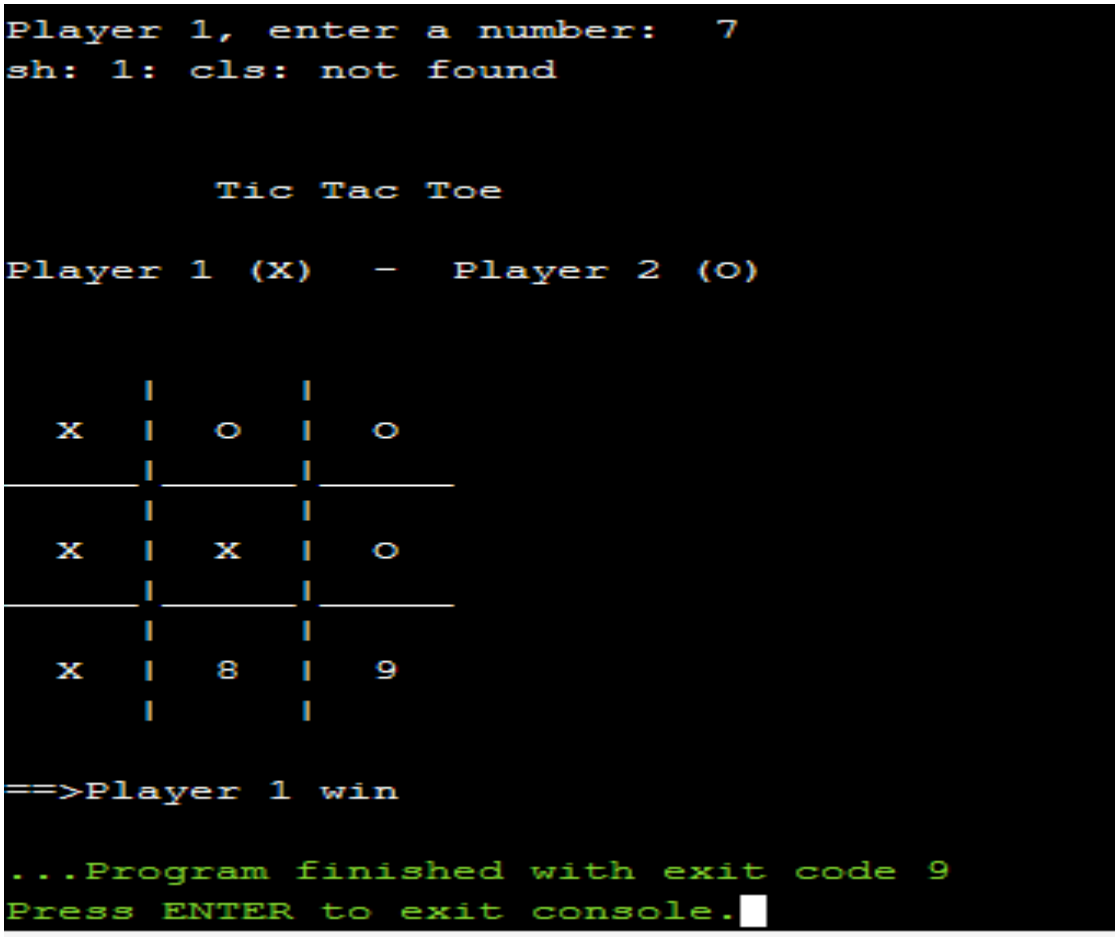

Player 2, enter a number: 3
sh: 1: cls: not found
Tic Tac Toe
Player 1 (X) - Player 2 (O)
1 | 2 | O
4 | X | 6
7 | 8 | 9


Player 1, enter a number: 1
sh: 1: cls: not found
Tic Tac Toe
Player 1 (X) - Player 2 (O)
X | 2 | O
4 | X | 6
7 | 8 | 9


Player 2, enter a number: 2
sh: 1: cls: not found
Tic Tac Toe
Player 1 (X) - Player 2 (O)
X | O | O
4 | X | 6
7 | 8 | 9


Player 1, enter a number: 4
sh: 1: cls: not found
Tic Tac Toe
Player 1 (X) - Player 2 (O)
X | O | O
X | X | 6
7 | 8 | 9


Player 2, enter a number: 6
sh: 1: cls: not found
Tic Tac Toe
Player 1 (X) - Player 2 (O)
X | O | O
X | X | O
7 | 8 | 9


Player 1, enter a number: 7
sh: 1: cls: not found
Tic Tac Toe
Player 1 (X) - Player 2 (O)
X | O | O
X | X | O
X | 8 | 9
==>Player 1 win
...Program finished with exit code 9
Press ENTER to exit console.

Practical: 3

AIM: - Write a program for Water Jug Problem.

```
#include<bits/stdc++.h>
using namespace std;
int x;
int y;
void show(int a, int b);
int min(int w, int z)
{
if (w < z)
return w;
else
return z;
}
void show(int a, int b)
{
cout << setw(12) << a << setw(12) << b<<endl;
}
void s(int n)
{
int xq = 0, yq = 0;
int t;
cout << setw(15) <<"FIRST JUG"<< setw(15) <<"SECOND JUG"<<endl;
while (xq != n && yq!=n )
{
  if (xq == 0)
  {
    xq = x;
    show(xq, yq);
  }
  else if (yq == y)
  {
    yq = 0;
    show(xq, yq);
  }
  else
  {
    t = min(y - yq, xq);
    yq= yq + t;
    xq = xq - t;
    show(xq, yq);
      }
}
}
int main()
{
int n;
cout << "Enter the liters of water required out of the two jugs: ";
```

```
cin >> n;
cout << "Enter the capacity of the first jug: ";
cin >> x;
cout << "Enter the capacity of the second jug: ";
cin >> y;
if(n<x || n<y)
{ if(n%(__gcd(x,y))==0)
   s(n);
  else
  cout<<"This is not possible....\n";
}
else
  cout<<"This is not possible....\n";
}
```

Output:-