

AI Lab Assignment 01

Problem Statement: *Tic-Tac-Toe*

NAME: **Sahil Dattatray Mohite**

CLASS: TY(IT)

ROLLNO: **29**

BATCH: **02**

Part1

```
#include <iostream>
#include<bits/stdc++.h>
#include <string>
#include <vector>

using namespace std;

int main()
{
    vector<int> board = {2,0,0,1,1,0,0,1,2};

    int count1=0,count2=0;

    for(int i=0; i<board.size(); i++)
    {
        if(board[i] == 1)
            count1++;
        if(board[i] == 2)
            count2++;
    }

    if((count1 - count2) > 1 || (count2 - count1) > 1)
    {
        cout << "Invalid move";
        return 0;
    }
    else
        cout << "Valid move" << endl;

    long long score=0;
    int j=8;
    for(int i=0; i<board.size(); i++)
    {
        score+=board[i]*pow(3,j);
        j--;
    }
}
```

```
    cout << "Score is: "<< score;
}
```

Output:

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
PS C:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab> cd "c
:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe\" ; if ($?) { g++ Part1.cpp -o Part1 } ; if ($?) { .\
Part1 }
Valid move
Score is: 13451
PS C:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe>
```

The screenshot shows the Visual Studio Code editor with two tabs: 'Part1.cpp' and 'Part2.cpp'. The 'Part1.cpp' tab is active, displaying the following C++ code:

```
Tic-Tac-Toe > C++ Part1.cpp > ...
1 #include <iostream>
2 #include<bits/stdc++.h>
3 #include <string>
4 #include <vector>
5
6 using namespace std;
7
8 int main()
9 {
10     vector<int> board = {2,0,0,1,1,0,0,1,2};
11
12     int count1=0,count2=0;
13
14     for(int i=0; i<board.size(); i++)
15     {
16         if(board[i] == 1)
17             count1++;
18         if(board[i] == 2)
19             count2++;
20     }
21
22     if((count1 - count2) > 1 || (count2 - count1) > 1)
23     {
24         cout << "Invalid move";
```

The PowerShell terminal on the right shows the output of the command execution:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab> cd "c
:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe\" ; if ($?) { g++ Part1.cpp -o Part1 } ; if ($?) { .\
Part1 }
Valid move
Score is: 13451
PS C:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe>
```

Part2

```
#include <iostream>
#include<bits/stdc++.h>
#include <string>
#include <vector>

using namespace std;

int isAlreadyWin(vector<int> &pos){

    int i=0;
    int n=0;

    //row-wise
    while(n!=3)
    {
        if(pos[i]==pos[i+1] && pos[i+1]==pos[i+2] && pos[i] == 1)
        {
            return 0;
        }
        n++;
        i+=3;
    }

    n=i=0;

    //column-wise
    while(n!=3)
    {
        if(pos[i]==pos[i+3] && pos[i+3]==pos[i+6] && pos[i] == 1)
        {
            return 0;
        }
        n++;
        i++;
    }

    //checking both diagonals
    if(pos[0]==pos[4] && pos[4]==pos[8] && pos[0] == 1)
    {
        return 0;
    }

    if(pos[2]==pos[4] && pos[4]==pos[6] && pos[2] ==1)
    {
        return 0;
    }
}
```

```

        return 1;
    }

int isWin(vector<int> &pos){

    int i=0;
    int n=0;

    //row-wise
    while(n!=3)
    {
        if(pos[i]==pos[i+1] && pos[i+1]==pos[i+2] && pos[i] != 0)
        {
            return 100;
        }
        n++;
        i+=3;
    }

    n=i=0;

    //column-wise
    while(n!=3)
    {
        if(pos[i]==pos[i+3] && pos[i+3]==pos[i+6] && pos[i] != 0)
        {
            return 100;
        }
        n++;
        i++;
    }

    //checking both diagonals
    if(pos[0]==pos[4] && pos[4]==pos[8] && pos[0] != 0)
    {
        return 100;
    }

    if(pos[2]==pos[4] && pos[4]==pos[6] && pos[2] !=0)
    {
        return 100;
    }

    return 0;
}

int isBlocking(vector<int> &pos, vector<int> &board){

```

```

unordered_map<int,int> mp;

//checking degree
for(int i=0; i<9; i++)
{
    if(i%2==1)
    {
        mp[i]=2;
    }
    else if(i%2==0 && i!=4)
    {
        mp[i]=3;
    }
    else
    {
        mp[i]=4;
    }
}

for(int i=0; i<9; i++)
{
    //checking where we have inserted 2
    if(board[i]==0 && pos[i]==2){
        return mp[i];
    }
}

return 0;
}

int isPlayerWin(vector<int> &pos){

    int n=0, i=0;
    int count1=0, count2=0;

    //checking all rows
    while(n!=3)
    {
        count1=count2=0;
        for(int j=0;j<3;j++)
        {
            if(pos[i]==1)
            {
                count1++;
            }
            if(pos[i]==2)
            {
                count2++;
            }
        }
        n++;
    }
}

```

```

        }
        i++;
    }
    n++;
    if(count1==2 && count2==1)
    {
        return 50;
    }
}

```

```

n=0, i=0;
count1=0, count2=0;

```

```

//checking all columns
while(n!=3)
{
    count1=count2=0;
    i=n;
    for(int j=0; j<3; j++)
    {
        if(pos[i]==1){
            count1++;
        }
        if(pos[i]==2){
            count2++;
        }

        i=i+3;
    }

    n++;
    if(count1==2 && count2==1)
    {
        return 50;
    }
}

```

```

// if(count1==2 && count2==1)
// {
//     return 50;
// }

```

```

//checking cross diagonal - (2-4-6)
i=2;
count1=0,count2=0;

```

```

        for(int j=0; j<3; j++)
        {
            if(pos[i]==1)
            {
                count1++;
            }
            if(pos[i]==2)
            {
                count2++;
            }
            i=i+2;
        }

        if(count1==2 && count2==1)
        {
            return 50;
        }

        //checking cross diagonal - (0-4-8)
        i=0,count1=0,count2=0;

        for(int j=0; j<3; j++)
        {
            if(pos[i]==1)
            {
                count1++;
            }
            if(pos[i]==2)
            {
                count2++;
            }
            i=i+4;
        }

        if(count1==2 && count2==1)
        {
            return 50;
        }

        return 0;
    }

int main()
{
    vector<int> board = {0,0,1,2,0,0,0,0,0};
    //winning - 1,2,0,0,2,1,1,0,0
    //blocking - 1,1,0,2,0,0,0,0,0

```

```

//neither - 0,0,1,2,0,0,0,0,0

if(!isAlreadyWin(board)){
    cout<<"Game is over";
    return 0;
}

int count0=0;
int n=9;
int countm=0, k=0;

//Calculating empty positions
for(int i=0; i<n; i++)
{
    if(board[i]==0){
        count0++;
    }
}

vector<vector<int>> moves;

for(int i=0; i<count0; i++)
{
    moves.push_back(board);
}

for(int i=0; i<count0;i++)
{
    countm=0;

    //generating each possible move
    for(int j=0; j<n; j++)
    {
        if(moves[i][j]==0 && k==countm)
        {
            moves[i][j]=2; //computer is playing here
            k++;
            break;
        }
        else
        {
            if(moves[i][j]==0)
            {
                countm++;
            }
        }
    }
}
}

```



```

cout<<"Number of Possible Moves are : "<<count0<<endl<<endl;

for(int i=0; i<count0; i++)
{
    for(int j=0; j<n; j++)
    {
        cout<<moves[i][j]<<" ";
    }
    cout<<endl;
}

cout<<endl<<endl;

//Maintain score of each possible move generated
vector<int> score(count0,0);

for(int i=0; i<count0; i++)
{
    score[i]=isWin(moves[i]);

    if(score[i]==0)
    {
        score[i]=isPlayerWin(moves[i]);
    }
    if(score[i]==0)
    {
        score[i]=isBlocking(moves[i],board);
    }
}

cout<<"Scores of all Moves are : "<<endl;
for(int i=0;i<count0;i++)
{
    cout<<i+1<<")Move "<<i+1<<" score is "<<score[i]<<endl;
}

int g=0;
for(int i=0;i<count0;i++)
{
    if(score[i]>score[g]){
        g=i;
    }
}

g++;
cout<<endl<<endl<<"The Best Possible Move is "<<g<<endl;
return 0;

```

```

}

/*

Game Grid :

+---+---+---+
| 0 | 1 | 2 |
+---+---+---+
| 3 | 4 | 5 |
+---+---+---+
| 6 | 7 | 8 |
+---+---+---+

Indicators:

0 - Empty Spaces
1 - Player 1 (me)
2 - Player 2 (computer)

Think as you are computer. You have to code in such way that computer has to
win
*/

```

Output:

```

:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe\" ; if ($?) { g++ Part2.cpp -o Part2 } ; if ($?) { .\
Part2 }
Number of Possible Moves are : 7

2 0 1 2 0 0 0 0 0
0 2 1 2 0 0 0 0 0
0 0 1 2 2 0 0 0 0
0 0 1 2 0 2 0 0 0
0 0 1 2 0 0 2 0 0
0 0 1 2 0 0 0 2 0
0 0 1 2 0 0 0 2 0
0 0 1 2 0 0 0 0 2

Scores of all Moves are :
1)Move 1 score is 3
2)Move 2 score is 2
3)Move 3 score is 4
4)Move 4 score is 2
5)Move 5 score is 3
6)Move 6 score is 2
7)Move 7 score is 3

The Best Possible Move is 3
PS C:\Users\Dell\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe>

```

```
File Edit Selection View Go Run Terminal Help
Part2.cpp - AI Lab - Visual Studio Code

C++ Part1.cpp C++ Part2.cpp x
Tic-Tac-Toe > C++ Part2.cpp > main()
301 //maintain score of each possible move generated
302 vector<int> score(count0,0);
303
304 for(int i=0; i<count0; i++)
305 {
306     score[i]=isWin(moves[i]);
307
308     if(score[i]==0)
309     {
310         score[i]=isPlayerWin(moves[i]);
311     }
312     if(score[i]==0)
313     {
314         score[i]=isBlocking(moves[i],board);
315     }
316 }
317
318 cout<<"Scores of all Moves are : "<<endl;
319 for(int i=0;i<count0;i++)
320 {
321     cout<<i+1<<"Move "<<i+1<<" score is "<<score[i]<<endl;
322 }
323
324 int g=0;
```

```

:Users\De11\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe\
e\" ; if ($?) { g++ Part2.cpp -o Part2 } ; if ($?) { .\
Part2 }
Number of Possible Moves are : 7

2 0 1 2 0 0 0 0 0
0 2 1 2 0 0 0 0 0
0 0 1 2 2 0 0 0 0
0 0 1 2 0 2 0 0 0
0 0 1 2 0 0 2 0 0
0 0 1 2 0 0 0 2 0
0 0 1 2 0 0 0 0 2

Scores of all Moves are :
1)Move 1 score is 3
2)Move 2 score is 2
3)Move 3 score is 4
4)Move 4 score is 2
5)Move 5 score is 3
6)Move 6 score is 2
7)Move 7 score is 3

The Best Possible Move is 3
PS C:\Users\De11\OneDrive\Desktop\VS Code\AI Lab\Tic-Tac-Toe>
```

Ln 318, Col 46 Spaces: 4 UTF-8 CRLF C++ Go Live Win32