Name : Nupur Suresh Shinde.

Class : TY-IT-A

Roll No : 54

Batch No : B2

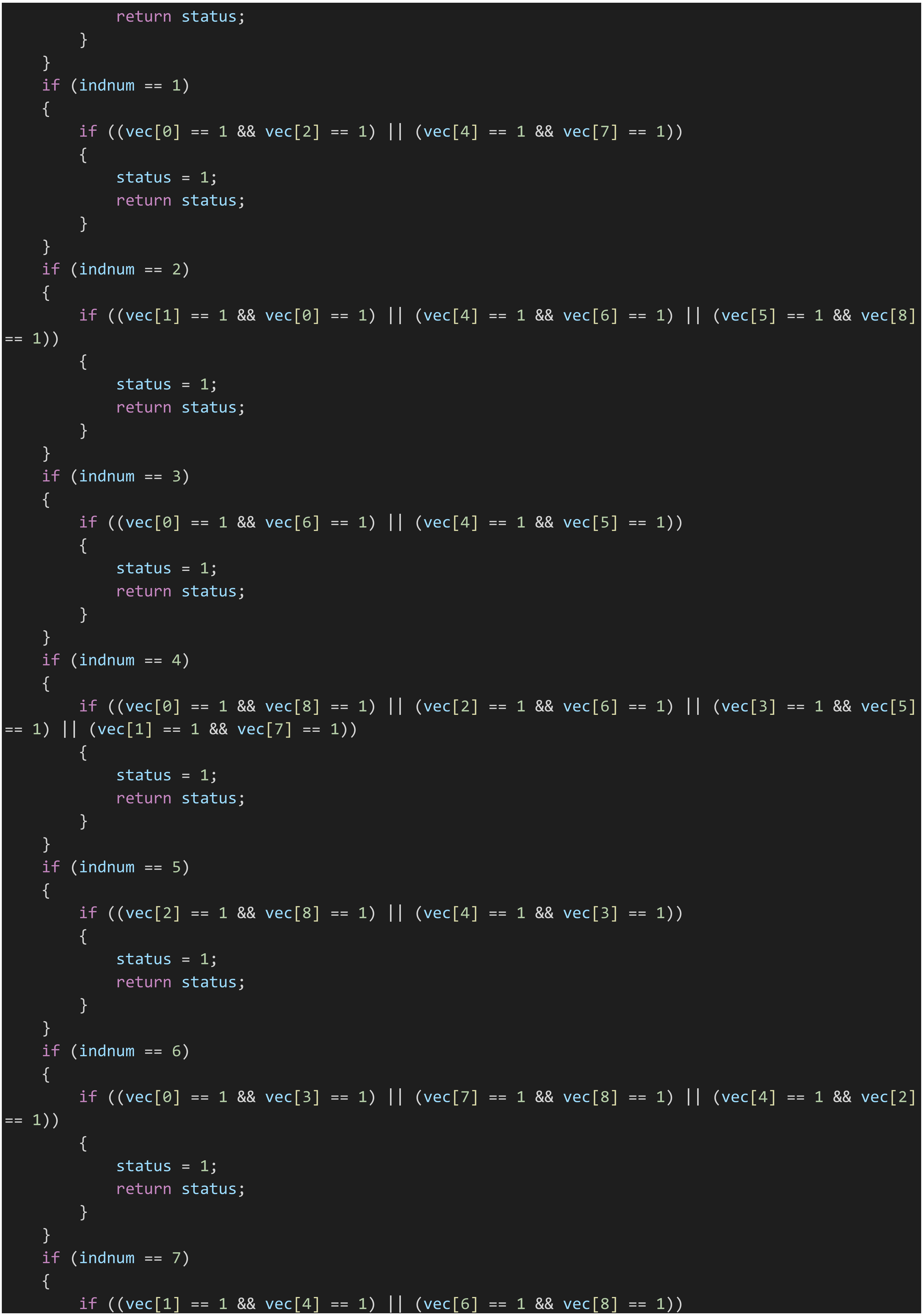
PRN No : 12010610

Lab No : 01 Part B

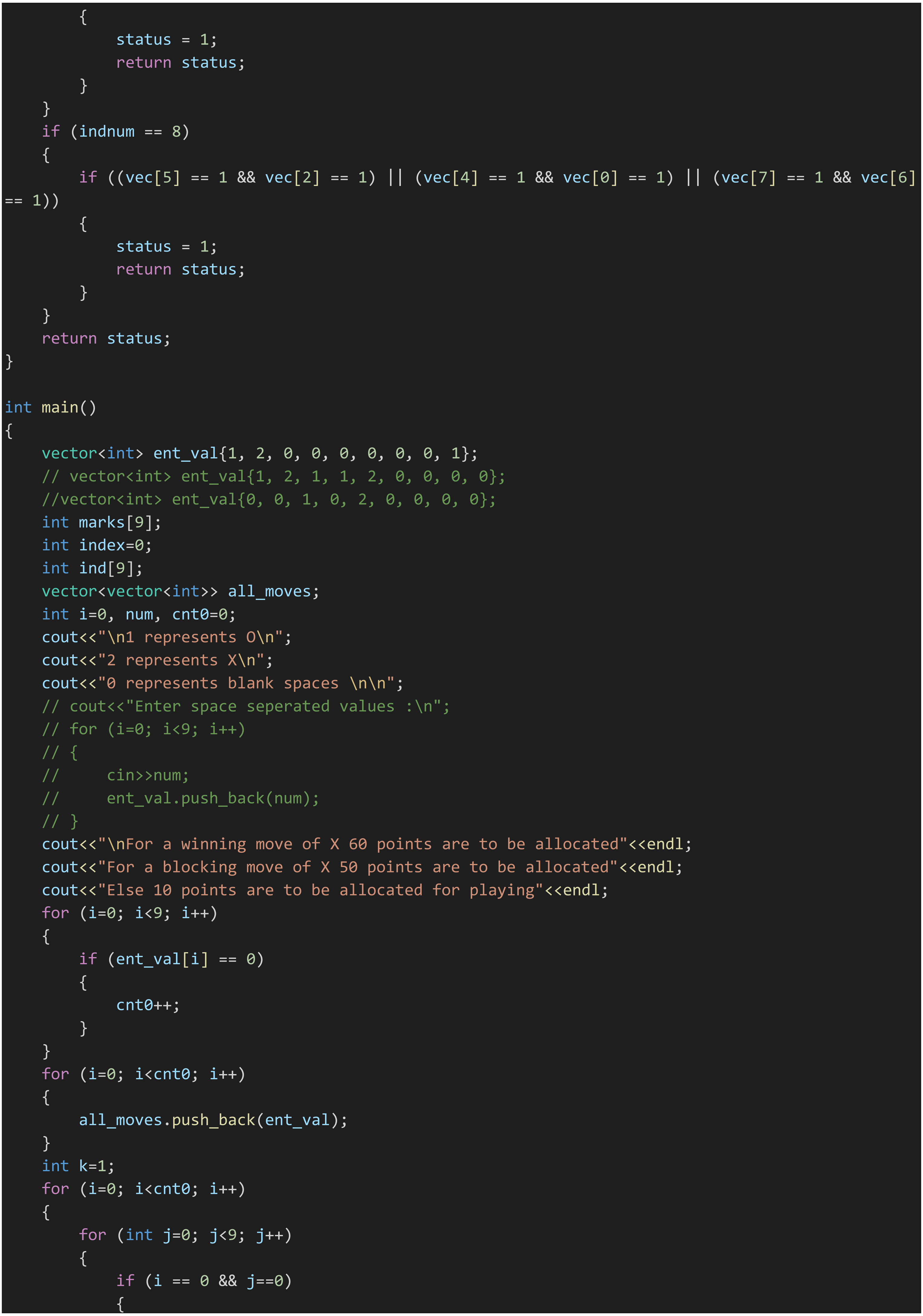
Problem Statement : Tic-Tac-Toe Game using AI Approach

Code:

```
#include <iostream>
#include <vector>
using namespace std;

int winstatus(vector<int> vec)
{
    int status2 = 2;
    if (vec[0] == 2)
    {
        if ((vec[1] == 2 && vec[2] == 2) || (vec[4] == 2 && vec[8] == 2) || (vec[3] == 2 && vec[6] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[1] == 2)
    {
        if ((vec[0] == 2 && vec[2] == 2) || (vec[4] == 2 && vec[7] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[2] == 2)
    {
        if ((vec[1] == 2 && vec[0] == 2) || (vec[4] == 2 && vec[6] == 2) || (vec[5] == 2 && vec[8] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[3] == 2)
    {
        if ((vec[0] == 2 && vec[6] == 2) || (vec[4] == 2 && vec[5] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[4] == 2)
    {
        if ((vec[0] == 2 && vec[8] == 2) || (vec[2] == 2 && vec[6] == 2) || (vec[3] == 2 && vec[5] == 2) || (vec[1] == 2 && vec[7] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[5] == 2)
    {
        if ((vec[2] == 2 && vec[8] == 2) || (vec[4] == 2 && vec[3] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[6] == 2)
    {
        if ((vec[0] == 2 && vec[3] == 2) || (vec[7] == 2 && vec[8] == 2) || (vec[4] == 2 && vec[2] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[7] == 2)
    {
        if ((vec[1] == 2 && vec[4] == 2) || (vec[6] == 2 && vec[8] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    if (vec[8] == 2)
    {
        if ((vec[5] == 2 && vec[2] == 2) || (vec[4] == 2 && vec[0] == 2) || (vec[7] == 2 && vec[6] == 2))
        {
            status2 = 1;
            return status2;
        }
    }
    return status2;
}

int blockstatus(vector<int> vec, int indnum)
{
    int status = 2;
    if (indnum == 0)
    {
        if ((vec[1] == 1 && vec[2] == 1) || (vec[4] == 1 && vec[8] == 1) || (vec[3] == 1 && vec[6] == 1))
        {
            status = 1;
```

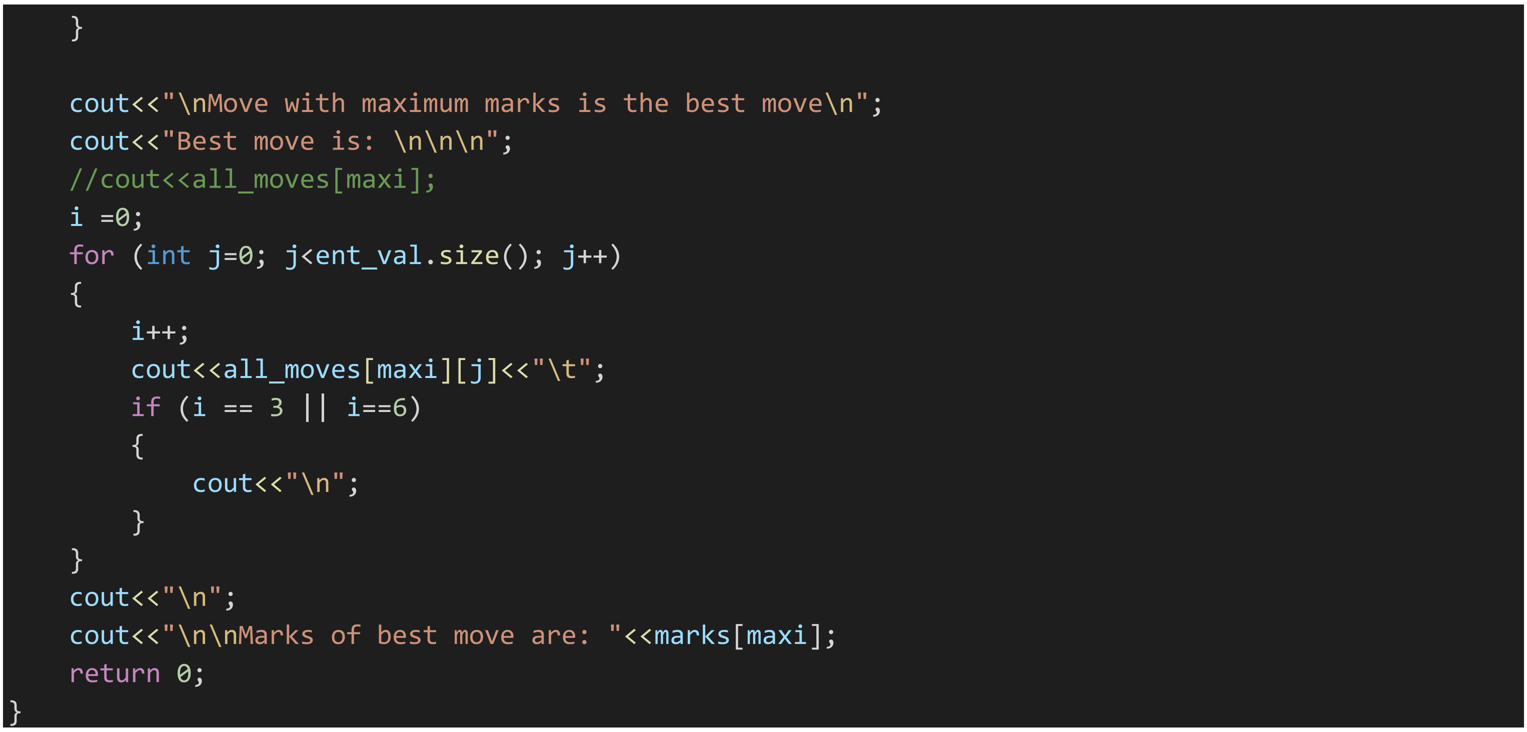


```
            return status;
        }
    }
    if (indnum == 1)
    {
        if ((vec[0] == 1 && vec[2] == 1) || (vec[4] == 1 && vec[7] == 1))
        {
            status = 1;
            return status;
        }
    }
    if (indnum == 2)
    {
        if ((vec[1] == 1 && vec[0] == 1) || (vec[4] == 1 && vec[6] == 1) || (vec[5] == 1 && vec[8] == 1))
        {
            status = 1;
            return status;
        }
    }
    if (indnum == 3)
    {
        if ((vec[0] == 1 && vec[6] == 1) || (vec[4] == 1 && vec[5] == 1))
        {
            status = 1;
            return status;
        }
    }
    if (indnum == 4)
    {
        if ((vec[0] == 1 && vec[8] == 1) || (vec[2] == 1 && vec[6] == 1) || (vec[3] == 1 && vec[5] == 1) || (vec[1] == 1 && vec[7] == 1))
        {
            status = 1;
            return status;
        }
    }
    if (indnum == 5)
    {
        if ((vec[2] == 1 && vec[8] == 1) || (vec[4] == 1 && vec[3] == 1))
        {
            status = 1;
            return status;
        }
    }
    if (indnum == 6)
    {
        if ((vec[0] == 1 && vec[3] == 1) || (vec[7] == 1 && vec[8] == 1) || (vec[4] == 1 && vec[2] == 1))
        {
            status = 1;
            return status;
        }
    }
    if (indnum == 7)
    {
        if ((vec[1] == 1 && vec[4] == 1) || (vec[6] == 1 && vec[8] == 1))
```



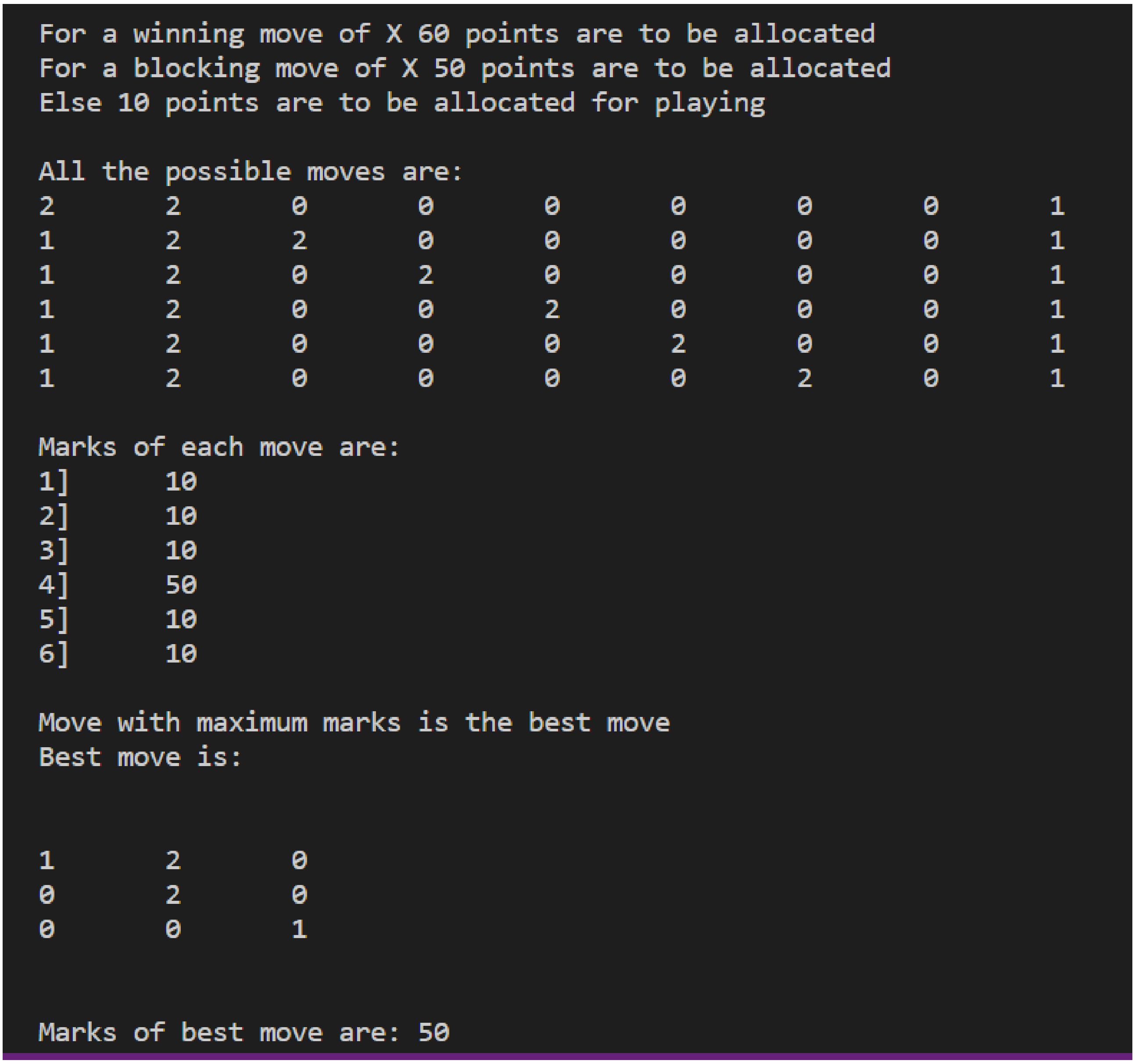
```
        {
            status = 1;
            return status;
        }
    }
    if (indnum == 8)
    {
        if ((vec[5] == 1 && vec[2] == 1) || (vec[4] == 1 && vec[0] == 1) || (vec[7] == 1 && vec[6]
== 1))
        {
            status = 1;
            return status;
        }
    }
    return status;
}

int main()
{
    vector<int> ent_val{1, 2, 0, 0, 0, 0, 0, 0, 1};
    // vector<int> ent_val{1, 2, 1, 1, 2, 0, 0, 0, 0};
    //vector<int> ent_val{0, 0, 1, 0, 2, 0, 0, 0, 0};
    int marks[9];
    int index=0;
    int ind[9];
    vector<vector<int>> all_moves;
    int i=0, num, cnt0=0;
    cout<<"\n1 represents O\n";
    cout<<"2 represents X\n";
    cout<<"0 represents blank spaces \n\n";
    // cout<<"Enter space seperated values :\n";
    // for (i=0; i<9; i++)
    // {
    //     cin>>num;
    //     ent_val.push_back(num);
    // }
    cout<<"\nFor a winning move of X 60 points are to be allocated"<<endl;
    cout<<"For a blocking move of X 50 points are to be allocated"<<endl;
    cout<<"Else 10 points are to be allocated for playing"<<endl;
    for (i=0; i<9; i++)
    {
        if (ent_val[i] == 0)
        {
            cnt0++;
        }
    }
    for (i=0; i<cnt0; i++)
    {
        all_moves.push_back(ent_val);
    }
    int k=1;
    for (i=0; i<cnt0; i++)
    {
        for (int j=0; j<9; j++)
        {
            if (i == 0 && j==0)
            {
```

```
                all_moves[0][0] = 2;
                break;
            }
            if(all_moves[i][j] == 0 && index < j)
            {
                all_moves[i][j]=2;
                index = j;
                ind[i] = index;
                //k = j;
                break;
            }
        }
    }
    for (int i=0; i<cnt0; i++)
    {
        int stat = winstatus(all_moves[i]);
        int indnum = ind[i];
        if (stat == 1)
        {
            marks[i] = 60;
        }
        else
        {
            int stat2 = blockstatus(all_moves[i], indnum);
            if (stat2 == 1)
            {
                marks[i] = 50;
            }
            else
            {
                marks[i] = 10;
            }
        }
    }
    cout<<"\nAll the possible moves are: \n";
    for (i=0; i<cnt0; i++)
    {
        for (int j=0; j<ent_val.size(); j++)
        {
            cout<<all_moves[i][j]<<"\t";
        }
        cout<<"\n";
    }
    cout<<"\nMarks of each move are: \n";
    for (i=0; i<cnt0; i++)
    {
        cout<<i+1<<"]"<<"\t"<<marks[i]<<endl;
    }
    int max = 0;
    int maxi = 0;
    for (int i=0; i<cnt0; i++)
    {
        if (marks[i] > max)
        {
            max = marks[i];
            maxi = i;
        }
```

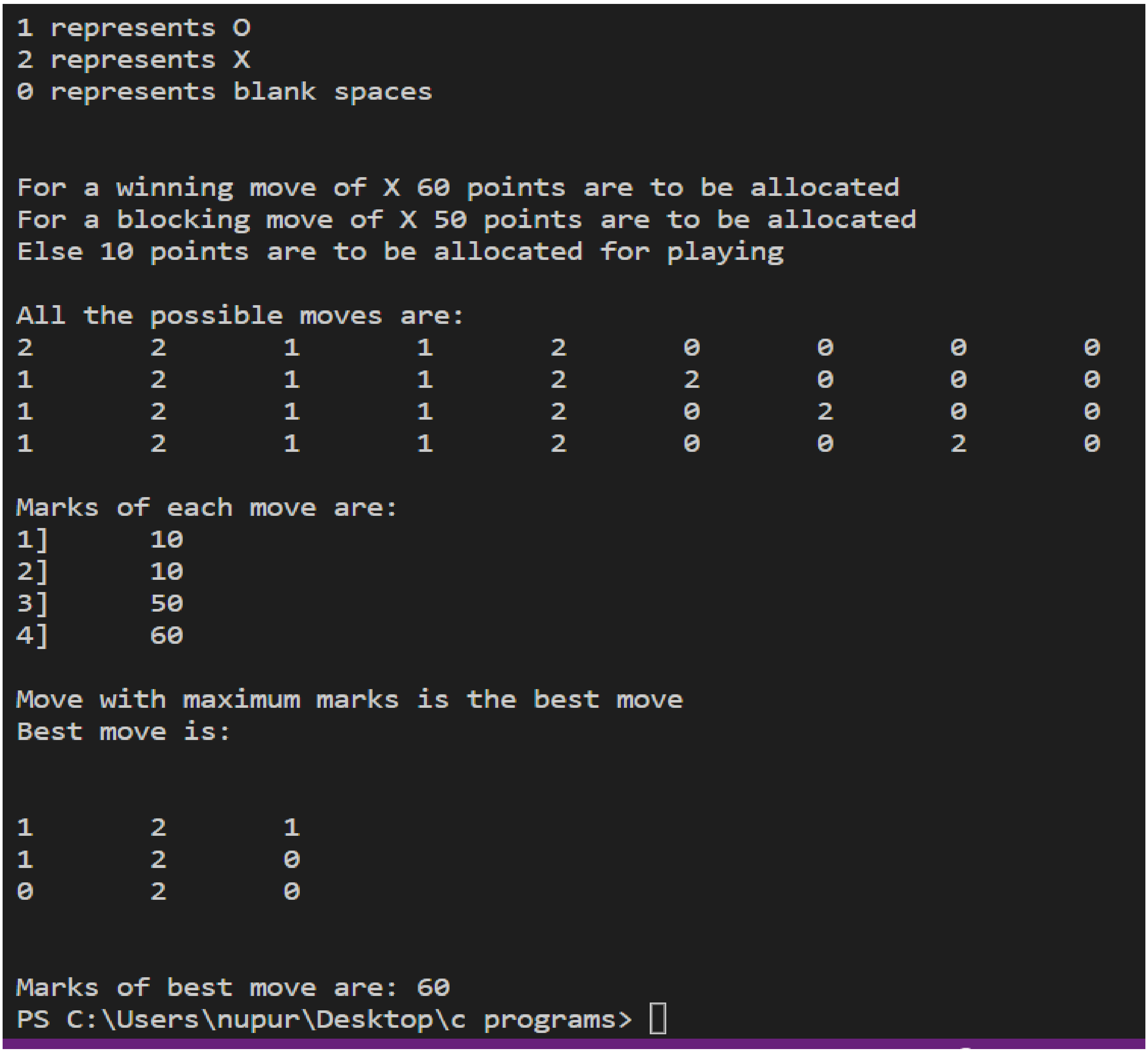


```
    }

    cout<<"\nMove with maximum marks is the best move\n";
    cout<<"Best move is: \n\n\n";
    //cout<<all_moves[maxi];
    i =0;
    for (int j=0; j<ent_val.size(); j++)
    {
        i++;
        cout<<all_moves[maxi][j]<<"\t";
        if (i == 3 || i==6)
        {
            cout<<"\n";
        }
    }
    cout<<"\n";
    cout<<"\n\nMarks of best move are: "<<marks[maxi];
    return 0;
}
```

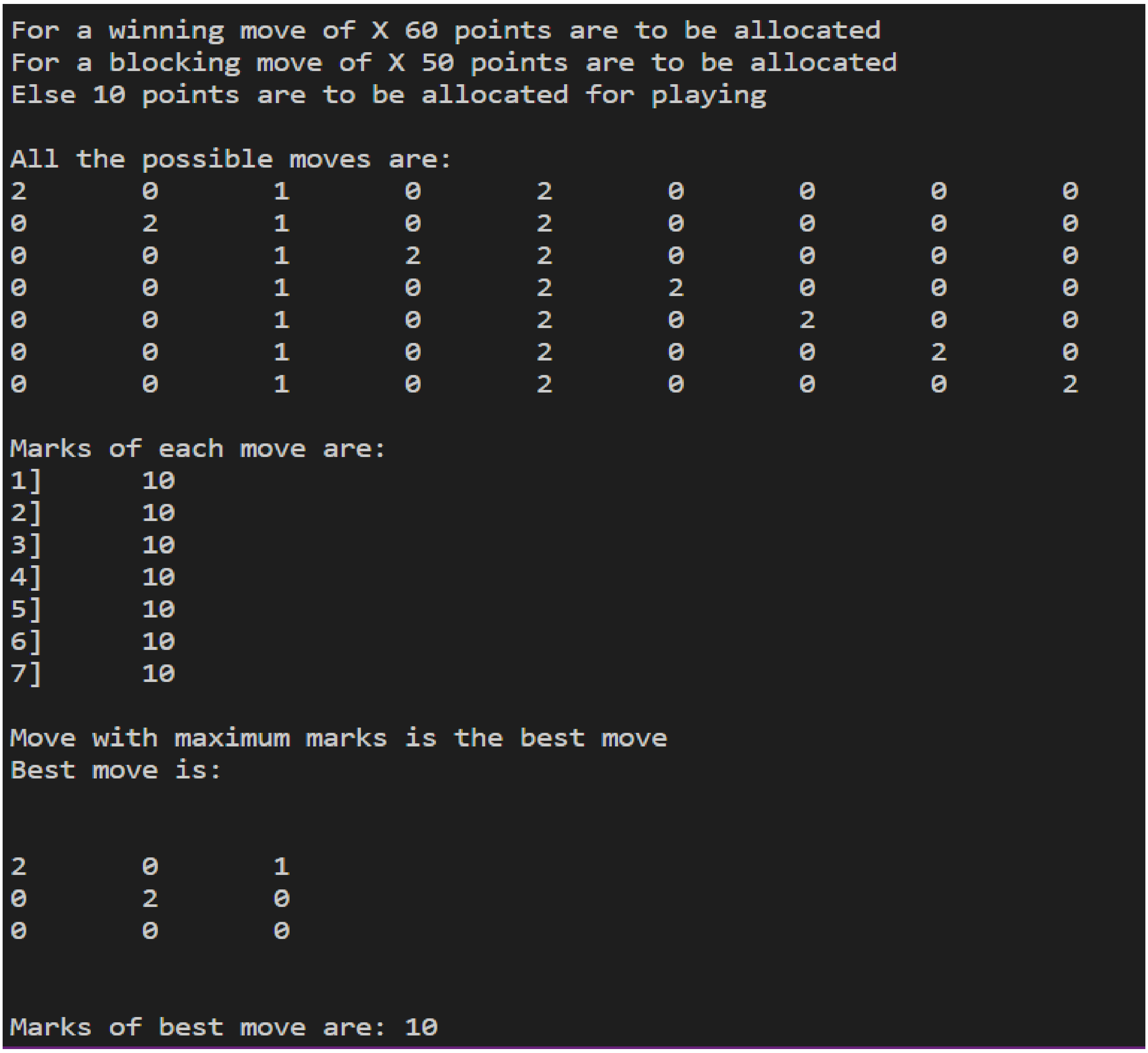
Output:



```
For a winning move of X 60 points are to be allocated
For a blocking move of X 50 points are to be allocated
Else 10 points are to be allocated for playing

All the possible moves are:
2       2       0       0       0       0       0       0       1
1       2       2       0       0       0       0       0       1
1       2       0       2       0       0       0       0       1
1       2       0       0       2       0       0       0       1
1       2       0       0       0       2       0       0       1
1       2       0       0       0       0       2       0       1

Marks of each move are:
1]      10
2]      10
3]      10
4]      50
5]      10
6]      10

Move with maximum marks is the best move
Best move is:


1       2       0
0       2       0
0       0       1


Marks of best move are: 50
```



```
1 represents O
2 represents X
0 represents blank spaces


For a winning move of X 60 points are to be allocated
For a blocking move of X 50 points are to be allocated
Else 10 points are to be allocated for playing

All the possible moves are:
2       2       1       1       2       0       0       0       0
1       2       1       1       2       2       0       0       0
1       2       1       1       2       0       2       0       0
1       2       1       1       2       0       0       2       0

Marks of each move are:
1]      10
2]      10
3]      50
4]      60

Move with maximum marks is the best move
Best move is:


1       2       1
1       2       0
0       2       0


Marks of best move are: 60
PS C:\Users\nupur\Desktop\c programs> 
```



```
For a winning move of X 60 points are to be allocated
For a blocking move of X 50 points are to be allocated
Else 10 points are to be allocated for playing

All the possible moves are:
2       0       1       0       2       0       0       0       0
0       2       1       0       2       0       0       0       0
0       0       1       2       2       0       0       0       0
0       0       1       0       2       2       0       0       0
0       0       1       0       2       0       2       0       0
0       0       1       0       2       0       0       2       0
0       0       1       0       2       0       0       0       2

Marks of each move are:
1]      10
2]      10
3]      10
4]      10
5]      10
6]      10
7]      10

Move with maximum marks is the best move
Best move is:


2       0       1
0       2       0
0       0       0


Marks of best move are: 10
```