Name- Jay S Maniyar

GR No- 1710982

Class – TY D

Roll No -7

**AI Home Assignment 2**

Q] Implementing Tic Tac Toe using **minmax** AI Heuristic

**Code**-

#include <iostream>

#include<conio.h>

using namespace std;

int cnt(int \*\*l, int n)

{

int m[3][3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

if (l[i][j] != 0)

m[i][j] = l[i][j];

else

m[i][j] = n;

}

}

int s = 0, t = 0, cnt = 0;

for (int i = 0; i < 3; i++)

{

s = 0;

t = 0;

for (int j = 0; j < 3; j++)

{

s = m[i][j] + s;

t = m[j][i] + t;

}

if (s == n \* 3)

cnt++;

if (t == n \* 3)

cnt++;

}

s = m[0][0] + m[1][1] + m[2][2];

t = m[0][2] + m[1][1] + m[2][0];

if (s == n \* 3)

cnt++;

if (t == n \* 3)

cnt++;

return cnt;

}

int winner (int \*\*l) //declaring the winner function

{

int s = 0, t = 0;

for (int i = 0; i < 3; i++)

{

s = 0;

t = 0;

for (int j = 0; j < 3; j++)

{

s = l[i][j] + s;

t = t + l[j][i];

}

//final Score for user to win is 300 while for the computer to win is 30

if (s == 300)

return 2; //Human Wins

else if (s == 30)

return 1; //Computer Wins

if (t == 300)

return 2; //Human Wins

else if (t == 30)

return 1; //Computer Wins

}

//Check First diagonal

s = l[1][1] + l[0][0] + l[2][2];

if (s == 300)

return 2; //Human Wins

else if (s == 30)

return 1; //Computer Wins

//Checking Second diagonal condition

s = l[1][1] + l[0][2] + l[2][0];

if (s == 300)

return 2; //Human Wins

else if (s == 30)

return 1;

return 0;

}

void computer\_turn(int \*\*l)

{

int minimum = 8, p, q, s, t;

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

{

if (l[i][j] == 0)

{

l[i][j] = 10;

s = cnt(l, 10);

t = cnt(l, 100);

if (minimum > s - t)

{

p = i;

q = j;

minimum = s - t;

}

l[i][j] = 100;

s = winner(l);

if (s == 2)

{

l[i][j] = 10;

return;

}

l[i][j] = 0;

}

}

l[p][q] = 10;

}

void system\_turn(int \*\*l)

{

int s;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

//Check for Empty spaces present

if (l[i][j] == 0)

{

//Placing the Computer's defualt value as 100

l[i][j] = 100;

s = winner(l);

if (s == 2)

{

l[i][j] = 10;

return;

}

l[i][j] = 10;

s = winner(l);

if (s == 1)

{

l[i][j] = 10;

return;

}

l[i][j] = 0;

}

}

}

int maximum = 0, k, p, t, u, q;

for (int i = 0; i < 3; i++)

{

s = 0;

t = 0;

for (int j = 0; j < 3; j++)

{

if (l[i][j] == 0)

{

t++;

k = i;

p = j;

}

else

if (l[i][j] == 10)

s++;

}

if (t>1 && s >= 1)

{

l[k][p] = 10;

return;

}

}

for (int i = 0; i < 3; i++)

{

s = 0;

t = 0;

for (int j = 0; j < 3; j++)

{

if (l[j][i] == 0)

{

t++;

k = j;

p = i;

}

else

if (l[j][i] == 10)

s++;

}

if (t>1 && s >= 1)

{

l[k][p] = 10;

return;

}

}

computer\_turn(l);

}

void print(int \*\*l)

{

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

if (l[i][j] == 10)

cout << "|\_X\_| ";

else if (l[i][j] == 100)

cout << "|\_O\_| ";

else

cout << " ";

}

cout << "\n";

}

cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

}

int main()

{

//Allocation of each 1 D array for every 2 D array

int \*\*l = new int \*[3];

for (int i = 0; i < 3; i++)

l[i] = new int[3];

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

l[i][j] = 0;

//Default location for Computer to play is 1 row and 1 Column

l[1][1] = 10;

int p, q, s,flag;

print(l); //let Human do the Input Funtion

for (int i = 0; i < 4; i++)

{

flag=1;

while(1)

{

cout << "Enter the row number(Choices are 1-->0 2-->1 and 3-->2) : ";

cin >> p;

cout << "Enter the column Number (Choices are 1-->0 2-->1 and 3-->2) : ";

cin >> q;

if(l[p][q]==0 && p<3&& q<3&& p>=0&& q>=0)

break;

else

{

cout<<"\n\nError:- Location already used up or invalid \t Enter choice again\n\n";

}

}

l[p][q] = 100;//Setting the Default value for user as 100

s = winner(l);

if (s == 1)

{

cout << ": Unfortunately Computer Won \n\n";

break;

}

else if (s == 2)

{

cout << ": Great User Won \n\n";

break;

}

//Repeating Computer's Turn

system\_turn(l);

//Performing the Repetitive Check

s = winner(l);

if (s == 1)

{

cout << "\*\*\*\*\*\*\*\*Computer Won\*\*\*\*\*\*\*\*\n";

break;

}

else if (s == 2)

{

cout << "\*\*\*\*\*\*\*\* Human Won\*\*\*\*\*\*\*\*\n";

break;

}

print(l);

}

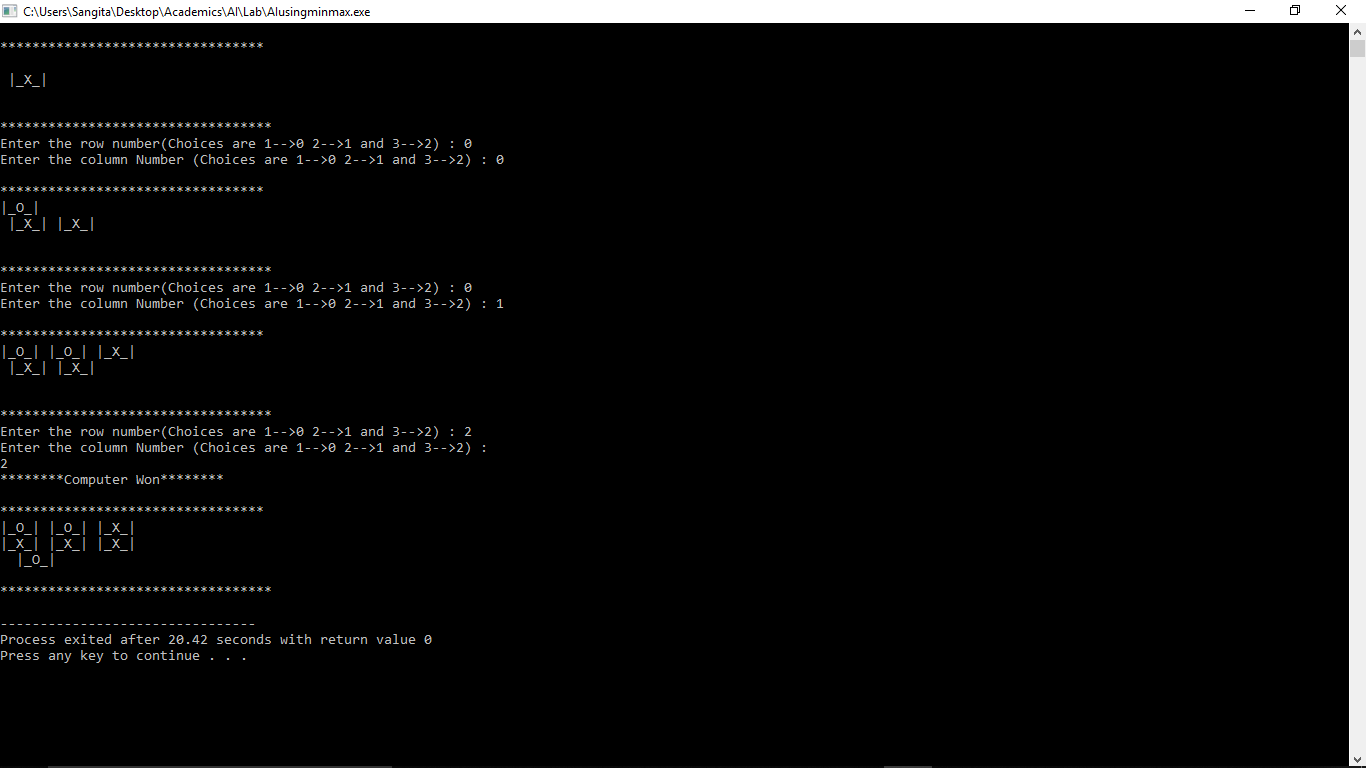
print(l);

return 0;

}

Output

Case 1] Computer Wins (Default Computer plays at 1 row and 1 Column)



Case II] Draw Condition

