Name : Nupur Suresh Shinde.

Class : TY-IT-A

Roll No : 54

Batch No : B2

PRN No : 12010610

Lab No : 02 Part A

Problem Statement : Calculating heuristic value of moves of 8 Puzzle game

Code:

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

bool issafe(int a, int b)
{
    if((a>=0 && a<3) && (b>=0 && b<3))
    {
        return true;
    }
    else
    {
        return false;
    }
}

int main()
{
    //int board[3][3] = {1, 5, 8, 2, 0, 3, 4, 6, 7};
    //int board[3][3] = {1, 8, 7, 2, 6, 0, 4, 3, 5};
    int board[3][3] = {1, 4, 0, 2, 5, 3, 8, 6, 7};
    int pos1[3][3], pos2[3][3], pos3[3][3], pos4[3][3];
    int goal[3][3] = {1, 2, 3, 8, 0, 4, 7, 6, 5};
    int zi, zj;
    int i, j;
    int heu1=0, heu2=0, heu3=0, heu4=0;
    int heusum1=0, heusum2=0, heusum3=0, heusum4=0;
    // cout << "Enter the elements of 8 puzzle matrix:\n";
    // for (i = 0; i < 3; i++)
    // {
    //     for (j = 0; j < 3; j++)
    //     {
    //         cin >> board[i][j];
    //     }
    // }
    cout << "Elements of 8 puzzle matrix:\n";
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
```

```
        {
            cout << board[i][j] << "\t";
        }
        cout << "\n";
    }
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            if (board[i][j] == 0)
            {
                zi = i;
                zj = j;
                break;
            }
        }
    }
    cout<<"All possible moves are:\n";
    if(issafe(zi-1, zj))
    {
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                pos1[i][j] = board[i][j];
            }
        }
        pos1[zi][zj] = pos1[zi-1][zj];
        pos1[zi-1][zj] = 0;
        cout<<"\nFirst possible move:\n";
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                cout<<pos1[i][j]<<"\t";
            }
            cout<<"\n";
        }
        int sqr;
        for(int i=0; i<3; i++)
        {
            for (int j=0; j<3; j++)
            {
                if(pos1[i][j] != goal[i][j])
                {
                    heu1++;
                }
                int ans1 = pos1[i][j]-goal[i][j];
                int sq = ans1*ans1;
                heusum1 = heusum1+sq;
                //sqr = sqrt(heusum1);
            }
        }
        int ans = sqrt(heusum1);
        cout<<"Heuristic value of first move: "<<heu1<<endl;
        cout<<"Heuristic value of first move according to distance: "<<ans<<endl;
    }
```

```
    if(issafe(zi+1, zj))
    {
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                pos2[i][j] = board[i][j];
            }
        }
        pos2[zi][zj] = pos2[zi+1][zj];
        pos2[zi+1][zj] = 0;
        cout<<"\nSecond possible move:\n";
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                cout<<pos2[i][j]<<"\t";
            }
            cout<<"\n";
        }
        for(int i=0; i<3; i++)
        {
            for (int j=0; j<3; j++)
            {
                if(pos2[i][j] != goal[i][j])
                {
                    heu2++;
                }
                int ans2 = pos2[i][j] - goal[i][j];
                int sqe = ans2 * ans2;
                heusum2 = heusum2+sqe;
            }
        }
        int an = sqrt(heusum2);
        cout<<"Heuristic value of second move: "<<heu2<<endl;
        cout<<"Heuristic value of second move according to distance: "<<an<<endl;
    }
    if(issafe(zi, zj-1))
    {
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                pos3[i][j] = board[i][j];
            }
        }
        pos3[zi][zj] = pos3[zi][zj-1];
        pos3[zi][zj-1] = 0;
        cout<<"\nThird possible move:\n";
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                cout<<pos3[i][j]<<"\t";
            }
            cout<<"\n";
        }
```

```
        for(int i=0; i<3; i++)
        {
            for (int j=0; j<3; j++)
            {
                if(pos3[i][j] != goal[i][j])
                {
                    heu3++;
                }
                int ans3 = pos3[i][j] - goal[i][j];
                int sqq = ans3 * ans3;
                heusum3 = heusum3 + sqq;
            }
        }
        int answ = sqrt(heusum3);
        cout<<"Heuristic value of third move: "<<heu3<<endl;
        cout<<"Heuristic value of third move according to distance: "<<answ<<endl;
    }
    if(issafe(zi, zj+1))
    {
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                pos4[i][j] = board[i][j];
            }
        }
        pos4[zi][zj] = pos4[zi][zj+1];
        pos4[zi][zj+1] = 0;
        cout<<"\nFourth possible move:\n";
        for(i=0; i<3; i++)
        {
            for(j=0; j<3; j++)
            {
                cout<<pos4[i][j]<<"\t";
            }
            cout<<"\n";
        }
        for(int i=0; i<3; i++)
        {
            for (int j=0; j<3; j++)
            {
                if(pos4[i][j] != goal[i][j])
                {
                    heu4++;
                }
                int ans4 = pos4[i][j] - goal[i][j];
                int sqw = ans4*ans4;
                heusum4 = heusum4 + sqw;
            }
        }
        int anq = sqrt(heusum4);
        cout<<"Heuristic value of fourth move: "<<heu4<<endl;
        cout<<"Heuristic value of fourth move according to distance: "<<anq<<endl;
    }
    return 0;
}
```

Output:

```
First possible move:
1       0       8
2       5       3
4       6       7
Heuristic value of first move: 7
Heuristic value of first move according to distance: 10

Second possible move:
1       5       8
2       6       3
4       0       7
Heuristic value of second move: 8
Heuristic value of second move according to distance: 12

Third possible move:
1       5       8
0       2       3
4       6       7
Heuristic value of third move: 7
Heuristic value of third move according to distance: 10

Fourth possible move:
1       5       8
2       3       0
4       6       7
Heuristic value of fourth move: 7
Heuristic value of fourth move according to distance: 10
PS C:\Users\nupur\Desktop\c programs>
```

```
Elements of 8 puzzle matrix:
1       8       7
2       6       0
4       3       5
All possible moves are:

First possible move:
1       8       0
2       6       7
4       3       5
Heuristic value of first move: 7
Heuristic value of first move according to distance: 12

Second possible move:
1       8       7
2       6       5
4       3       0
Heuristic value of second move: 8
Heuristic value of second move according to distance: 12

Third possible move:
1       8       7
2       0       6
4       3       5
Heuristic value of third move: 6
Heuristic value of third move according to distance: 10
PS C:\Users\nupur\Desktop\c programs>
```

```
Elements of 8 puzzle matrix:
1       4       0
2       5       3
8       6       7
All possible moves are:

Second possible move:
1       4       3
2       5       0
8       6       7
Heuristic value of second move: 6
Heuristic value of second move according to distance: 9

Third possible move:
1       0       4
2       5       3
8       6       7
Heuristic value of third move: 7
Heuristic value of third move according to distance: 8
PS C:\Users\nupur\Desktop\c programs>
```