**Code:**

#include <bits/stdc++.h>

using namespace std;

const char UP ='1';

const char DOWN= '2';

const char LEFT= '3';

const char RIGHT= '4';

void printArray(int\*\* array)

{

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

{

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

{

cout<< setw(8) << array[a][b];

}

cout<<endl;

}

cout<<endl;

}

// Calculate Manhattan distance

int manhattan\_distance(int\*\* start\_state,int\*\* goal\_state)

{

int manhattan\_distance=0;

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

{

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

{

if(start\_state[i][j] > 0)

{

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

{

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

{

if (goal\_state[k][l] == start\_state[i][j])

{

manhattan\_distance=manhattan\_distance + (abs(i-k)+abs(j-l));

}

}

}

}

}

}

return manhattan\_distance;

}

void makeMove(int\*\* temp,int move)

{

int flag=0;

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

{

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

{

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

{

if(move==1)

{

temp[i][j] = temp[i-1][j];

temp[i-1][j] = 0;

flag=1;

break;

}

else if(move==2)

{

temp[i][j] = temp[i+1][j];

temp[i+1][j] = 0;

flag=1;

break;

}

else if(move==3)

{

temp[i][j] = temp[i][j-1];

temp[i][j-1] = 0;

flag=1;

break;

}

else

{

temp[i][j] = temp[i][j+1];

temp[i][j+1] = 0;

flag=1;

break;

}

}

}

if(flag==1){break;}

}

}

//-----------------------------------------------------------------------------------------------------------

int tile\_Ordering(int\*\* current\_state,int\*\* goal\_state,int move)

{

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

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

{

temp[i]=new int[3]; // these represents columns

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

{

temp[i][j] = current\_state[i][j];

}

}

makeMove(temp,move);

printArray(temp);

int manhattan=manhattan\_distance(temp,goal\_state);

cout<<"Current Manhattan number :"<<manhattan<<endl<<endl<<endl;

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

{

delete temp[i];

}

delete temp;

return manhattan;

}

void steepestAscentHillClimbing( int\*\* start\_state, int\*\* goal\_state,int former\_move)

{

int arr[4] = {100,100,100,100};

cout<<"--------------------------------------------------------------------------------"<<endl;

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

{

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

{

if (start\_state[i][j] == 0)

{

if(i>0 && former\_move!=2)

{

cout<<"Checking child (moving 0 up) "<<endl;

arr[0]=tile\_Ordering(start\_state,goal\_state,1);

}

if(i<2 && former\_move!=1)

{

cout<<"Checking child (moving 0 down) "<<endl;

arr[1]=tile\_Ordering(start\_state,goal\_state,2);

}

if(j>0 && former\_move!=4)

{

cout<<"Checking child (moving 0 left) "<<endl;

arr[2]=tile\_Ordering(start\_state,goal\_state,3);

}

if(j<2 && former\_move!=3)

{

cout<<"Checking child (moving 0 right) "<<endl;

arr[3]=tile\_Ordering(start\_state,goal\_state,4);

}

}

}

cout<<endl;

}

int localOptimum=99;

int index=0;

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

{

if(arr[i]<localOptimum)

{

localOptimum=arr[i];

index=i+1;

}

}

makeMove(start\_state,index);

cout<<"Next state = minimum Manhattan number :"<<endl;

printArray(start\_state);

if(localOptimum==0)

{

cout<<"goal state reached"<<endl;

return;

}

else

steepestAscentHillClimbing(start\_state,goal\_state,index);

}

int main()

{

int\*\* initial = new int\*[3]; // allocate an array of 3 int pointers - these represents rows

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

{

initial[i]=new int[3]; // these represents columns

}

int\*\* final = new int\*[3]; // allocate an array of 3 int pointers - these represents rows

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

{

final[i]=new int[3]; // these represents columns

}

int player\_Input;

cout << "Enter initial board configuration - 0 denotes empty position" << endl;

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

{

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

{

cout<<"Enter input A["<<i<<"]["<<j<<"]"<<endl;

cin>>player\_Input;

initial[i][j]=player\_Input;

}

}

cout<<"--------------------------------------------------------------------------------"<<endl;

cout << "Enter final board configuration - 0 denotes empty position" << endl;

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

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

{

cout<<"Enter input A["<<i<<"]["<<j<<"]"<<endl;

cin>>player\_Input;

final[i][j]=player\_Input;

}

cout<<"\n---------------------------Your initial matrix is-------------------------------\n"<<endl;

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

{

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

{

cout << setw(8) << initial[i][j];

}

cout<<endl;

}

cout<<"\n---------------------------Your final matrix is--------------------------------\n"<<endl;

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

{

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

{

cout << setw(8) << final[i][j];

}

cout<<endl;

}

cout<<"\n--------------------------------------------------------------------------------"<<endl;

cout<<"\n--------------------------------------------------------------------------------"<<endl;

cout<<"\nCalling Steepest Ascent Hill Climbing function\n"<<endl;

steepestAscentHillClimbing(initial,final,0);

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

{

delete initial[i];

delete final[i];

}

delete initial;

delete final;

return 0;

}

**Output:**













