Program 5

Implement the 8-puzzle problem using A* algorithm, using Heuristic function as Manhattan distance with depth not more the 3. If goal state is not reached within this limit, agent must report “NOSOLUTION”.

Program:

```
GoalNode=[[1,2,3],[4,5,6],[7,8,0]]
StartNode=[[8,2,3],[0,4,6],[7,5,1]]
temp = []
h1 = -1
h2 = 0

print("Given StartNode is: ",StartNode)

print("\n\n\t Given GoalNode is: ",GoalNode)

print("\n\n#######################################")

for i in range(len(StartNode)):
    for j in range (len(StartNode)):
        if StartNode[i][j] != GoalNode[i][j]:
            h1+=1
print("\n\n\t h1 : Number of misplaced tiles =>",h1)



'''
for i in StartNode:
    for j in i:
        print("StartNode",j)

print("#####################################")
for i in GoalNode:
    for j in i:
        print("GoalNode",j)
print("#####################################")
for i in range(len(StartNode)):
    for j in range (len(StartNode)):
        print("i is ",i,"j is :",j)'''

print("\n\n#######################################")

print("\n\nDistances of the tiles from their goal positions are: \n")
```

```
for i in r%nge(leñ(StartNode)):
    for j in range (ten(Sta rtNo.de ) ) :
        if (Sta rtNode [1] [ j ] --0) :


            If (I2oa1Nade [ 0] 6] == Sta r1ho.de [ i ] [ j ] ) :
                temp. append.(abs (:i -8) + abs ( j -.8) )
                p rent ( " \t" , temp)

            ellf ( Ooa1Node [0] [1 ] -= S tar t lJode. [1] [ j ] ) :
                temp.append.(abs(:i -8) + abs(j -1) )
                p rInt(" \t" ,temp)
            el if (Gpa1Node [ B] [2 ] -• StartNode [a:] [ y ] } :
                temp. append (.abs (1-8) + abs ( j -2) )
                p rInt (" If " , teirp)
            ellf (GéalNéde[7][0] ==StartNode[i][j)):
                temp. append (abs (i -1) + abs ( j -0) )
                p rInt(" \t" , temp)
            ellf (Goa1Node[1][1] == StartNode.[z.][y]}:
                temp. append(abs(1-1) + abs( j -1) )
                p rInt (" If " , teirp)
            ellf ( Ooa1Node [1 ] [ 2] -= StartlJode [1] [ j ] ) :
                temp. append (abs (i -1) + abs ( j -2) )
                qrent ( " \t" , temp)
            ellf (Coa1Node [2.] [ 0] -• StarTNode.[1.] [ j ] } :
                temp. append (abs (1-2) + abs ( j -0) )
                p rInt (" \1 " , teixp)
            ellf ( Ooa1Node [2 ] [1 ] == StartlJode [1] [ j ] ) :
                temp. append.(abs (i -2) + abs ( j -1 ) )
                p rent ( " \t" , temp)
            ellf (Cpa1Node [2 ] [2] -• StarTNode.[1:] [ j ] } :
                temp. append(abs (1-2) + abs ( j -2) )
                p rInt (" \1 " , teixp)
            else:
                prznt("Warning'!' This is foi 8-puzzle progiam..So, don't cross the array lidit."}
```

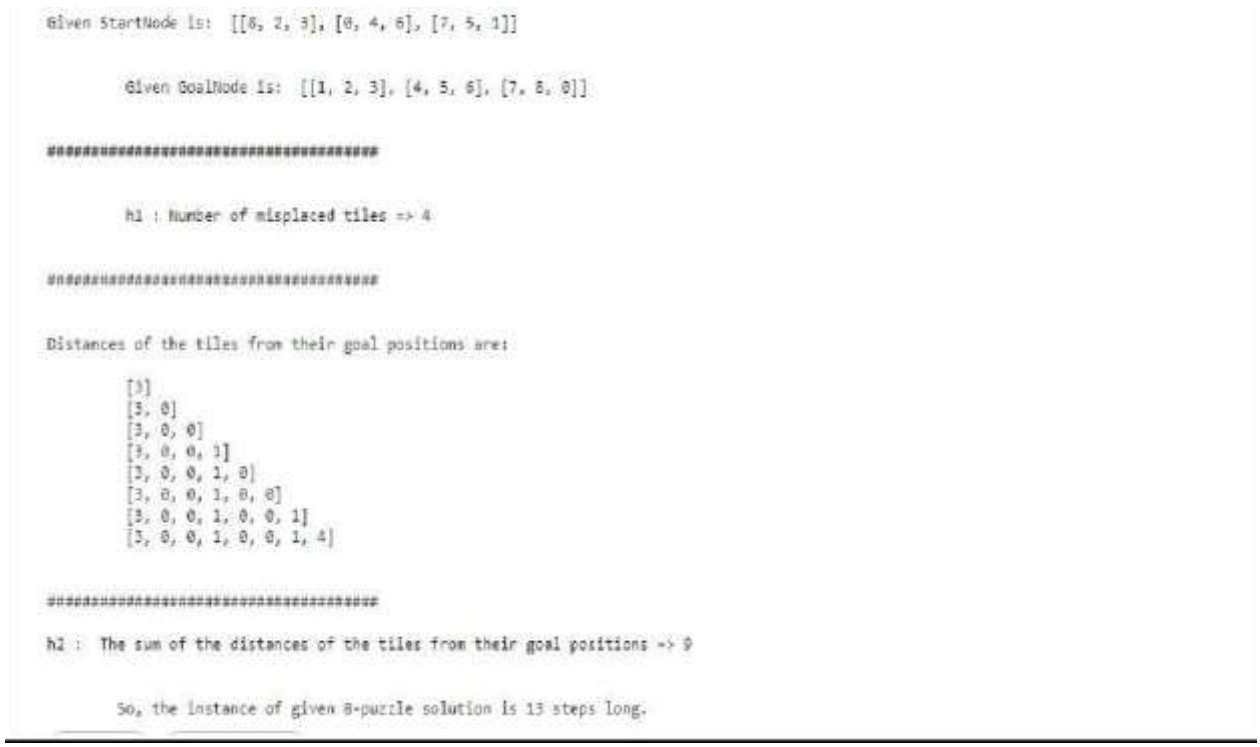
```
print("\n\n#####################################")

for i in range(len(temp)):
    h2+=temp[i]
print("\nh2 :  The sum of the distances of the tiles from their goal positions =>",h2)

h=h1+h2

print("\n\n\tSo, the instance of given 8-puzzle solution is",h,"steps long.")
```

Output:


Given StartNode is: [[8, 2, 3], [0, 4, 6], [7, 5, 1]]
Given GoalNode is: [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
h1 : Number of misplaced tiles => 4
Distances of the tiles from their goal positions are:
[3]
[3, 0]
[3, 0, 0]
[3, 0, 0, 1]
[3, 0, 0, 1, 0]
[3, 0, 0, 1, 0, 0]
[3, 0, 0, 1, 0, 0, 1]
[3, 0, 0, 1, 0, 0, 1, 4]
h2 : The sum of the distances of the tiles from their goal positions => 9
So, the instance of given 8-puzzle solution is 13 steps long.