

Program 4 Write UP

puzzle=[[-1 , 1 , 3] , [4 , 2 , 5] , [7 , 8 , 6]]

result=[[1 , 2 , 3] , [4 , 5 , 6] , [7 , 8 , -1]]

#Arr[] -> [[puzzle , gn , hn , action , fn]]

Def compare(puz)

Compare puz and result to find out h(n) value

Def findBlank()

Find blank (i.e. -1) in puzzle and return as (i , j)

Def copy(puz)

Return a duplicated array of puz

Def checkAll(Arr,depth,past)

Use findBlank to find (i , j)

**use i and j to calculate all the possible moves (except the pervious move
using past) and store the puzzle, hn, fn and action based on the
lowest fn value and then store it in Arr and then return Arr**

def aStar(Arr)

retrieve the last entry in Arr into temp

if hn value in temp == 0 then print the entire Arr array and return true

if gn value in temp >=15 then print the entire Arr and return false

use checkAll() by sending in Arr and the temp's-> gn and action

recursively call itself

def Depth()

create an Arr such that its first entry stores puzzle , gn (which is 0), hn

(compare(puzzle)), action (dummy action), fn (compare(puzzle)) in

the same order

call aStar

if it return false then print unsuccessful

else print successful

def main()

input the puzzle from user into puzzle

call depth()

main():