**Input :**

def misplaced\_tiles(state, goal\_state):

misplaced = 0

for i in range(3):

for j in range(3):

if state[i][j] != 0 and state[i][j] != goal\_state[i][j]:

misplaced += 1

return misplaced

def valid\_moves(state):

moves = []

zero\_pos = next((i, j) for i in range(3) for j in range(3) if state[i][j] == 0)

x, y = zero\_pos

directions = [(-1, 0), (1, 0), (0, -1), (0, 1)]

for dx, dy in directions:

new\_x, new\_y = x + dx, y + dy

if 0 <= new\_x < 3 and 0 <= new\_y < 3:

new\_state = [row[:] for row in state]

new\_state[x][y], new\_state[new\_x][new\_y] = new\_state[new\_x][new\_y], new\_state[x][y]

moves.append(new\_state)

return moves

def a\_star(initial\_state, goal\_state):

open\_list = [(initial\_state, [], 0)] # The list holds tuples of (state, path, g)

closed\_list = set()

while open\_list:

current\_state, path, g = open\_list.pop(0)

if current\_state == goal\_state:

return path

closed\_list.add(tuple(map(tuple, current\_state)))

for move in valid\_moves(current\_state):

if tuple(map(tuple, move)) not in closed\_list:

f = g + 1 + misplaced\_tiles(move, goal\_state)

open\_list.append((move, path + [move], g + 1))

open\_list.sort(key=lambda x: x[2] + misplaced\_tiles(x[0], goal\_state)) # Sort by f = g + h

return None

def print\_state(state):

for row in state:

print(row)

print()

# Get initial state and goal state from user

def get\_state\_input(prompt):

print(prompt)

state = []

for i in range(3):

row = list(map(int, input(f"Enter row {i + 1} (space-separated): ").split()))

state.append(row)

return state

initial\_state = get\_state\_input("Enter the initial state:")

goal\_state = get\_state\_input("Enter the goal state:")

print("\nStart State:")

print\_state(initial\_state)

print("h = ", misplaced\_tiles(initial\_state, goal\_state), "\n")

print("Goal State:")

print\_state(goal\_state)

solution = a\_star(initial\_state, goal\_state)

if solution:

print("Steps to reach the goal:")

for step in solution:

print\_state(step)

print("h = ", misplaced\_tiles(step, goal\_state), "\n")

else:

print("No solution found.")

**Output :**

admin1@admin1-ThinkCentre-E73:~$ python3 a\_star.py

Enter the initial state:

Enter row 1 (space-separated): 2 8 3

Enter row 2 (space-separated): 1 6 4

Enter row 3 (space-separated): 7 0 5

Enter the goal state:

Enter row 1 (space-separated): 1 2 3

Enter row 2 (space-separated): 8 0 4

Enter row 3 (space-separated): 7 6 5

Start State:

[2, 8, 3]

[1, 6, 4]

[7, 0, 5]

h = 4

Goal State:

[1, 2, 3]

[8, 0, 4]

[7, 6, 5]

Steps to reach the goal:

[2, 8, 3]

[1, 0, 4]

[7, 6, 5]

h = 3

[2, 0, 3]

[1, 8, 4]

[7, 6, 5]

h = 3

[0, 2, 3]

[1, 8, 4]

[7, 6, 5]

h = 2

[1, 2, 3]

[0, 8, 4]

[7, 6, 5]

h = 1

[1, 2, 3]

[8, 0, 4]

[7, 6, 5]

h = 0