PAGE NO :
DATE :

Code

```
import heapq
import numpy as np
q = []
goal = [[0,1,2],[3,4,5],[6,7,8]]
vis = set()
parent_map, move_map = {}, {}

def manhatten(curr):
    ans = 0
    pos = { goal[i][j] : (i,j) for i in range(3) for j in
                                                range(3) }

    for i in range(3):
        for j in range(3):
            x,y = pos[curr[i][j]]
            ans += abs(i-x) + abs(j-y)

    return ans

def moves(curr):
    x,y = [(i,j) for i in range(3) for j in range(3)
                        if curr[i][j]==0][0]
    poss = [[0,-1,'left'],[-1,0,'up'],[1,0,'down']
                        [0,1,'right']]

    for dx,dy,direction in poss:
        nx,ny = x+dx, y+dy
        if 0<=nx<3 and 0<=ny<3:
            curr1 = [row[:] for row in curr]
            curr1[x][y], curr1[nx][ny] = curr1[nx][ny],
                                        curr1[x][y]
            t_curr1 = tuple(map(tuple,curr1))
            if t_curr1 not in vis:
```

PAGE NO :
DATE :

```
        heapq.heappush(q,(manhatten(curr),curr))
        vis.add(t_curr)
        parent_map[tuple(map(tuple,curr))]=curr
        move_map[tuple(map(tuple,curr))]=direction

def dfs(curr):
    vis.add(tuple(map(tuple,curr))
    if curr==goal:
        return True
    moves(curr)
    if q:
        curr=heapq.heappush(q)[1]
        if dfs(curr):
            return True
    return False

def display(board):
    print("+---+---+---+")
    for row in board:
        print("|"+"|".join(str(x) if x!=0 else ' ' for x in row)+"|")
        print("+---+---+---+")

c=[[] for i in range(3)]
for i in range(3):
    print(f"Enter elements of row{i+1}")
    c[i]=list(map(int,input().split()))
dfs()

result_path=[]
directions=[]
state=goal
```

```
while state:
    result_path.append(state)
    direction.append(move_map.get(tuple(map(tuple,state))))
    state = parent_map.get(tuple(map(tuple,state)))

for ind, (state, direction) in enumerate(reversed(list(zip(
                                    result_path, direction)))):
    print(f"Step {ind}:")
    display(state)
    if ind == 0:
        print("Initial State")
    if direction:
        print(f"Move empty space {direction}")
    print()

print(f"Steps taken: {len(result_path)-1}")
```

Output :-

```
Enter elements of row 1: 3 7 6
Enter elements of row 2: 4 5 8
Enter elements of row 3: 2 0 1

Step 0:
+---+---+---+
| 3 | 7 | 6 |
+---+---+---+
| 2 | 4 | 8 |
+---+---+---+
| 2 |   |   |
+---+---+---+
Initial State
```

PAGE NO :
DATE :

Step 1 :

3	1	7	6
4		8	
2	5		

Move empty space up

:

Step 59 :

	1	1	2
3	4	5	
6	7	8	

Move empty space up

8

Steps taken : 59