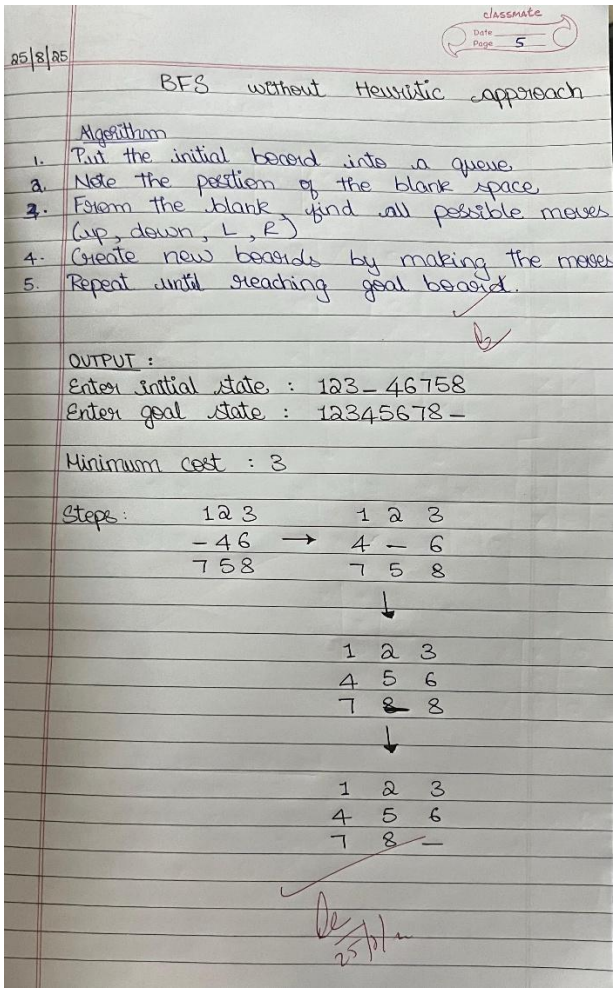
Lab3a (25.08.2025)

Sharada Koundinya(1BM23CS310)

Implement BFS without Heuristic approach




```
from collections import deque

def get_moves(state):
    idx = state.index("_")
    x, y = divmod(idx, 3)
    moves = []
    for dx, dy in [(-1,0),(1,0),(0,-1),(0,1)]:
        nx, ny = x+dx, y+dy
        if 0 <= nx < 3 and 0 <= ny < 3:
            nidx = nx*3 + ny
            lst = list(state)
```

```
            lst[idx], lst[nidx] = lst[nidx], lst[idx]
            moves.append("".join(lst))
    return moves

def bfs(start, goal):
    q = deque([(start, 0)])
    parent = {start: None}
    visited = {start}
    while q:
        state, cost = q.popleft()
        if state == goal:
            path = []
            while state:
                path.append(state)
                state = parent[state]
            path.reverse()
            return path, cost
        for move in get_moves(state):
            if move not in visited:
                visited.add(move)
                parent[move] = state
                q.append((move, cost+1))

start = input("Enter initial state (e.g., 54_618732): ")
goal = input("Enter goal state (e.g., 12345678_): ")
path, cost = bfs(start, goal)
print("Minimum cost:", cost)
print("Steps:")
for p in path:
```

```
    for i in range(0, 9, 3):
        print(p[i:i+3])


    print()
print("Sharada Koundinya,1BM23CS310")
```

Output:

```
Enter initial state (e.g., 54_618732): 123_46758
Enter goal state (e.g., 12345678_): 12345678_
Minimum cost: 3
Steps:
123
_46
758

123
4_6
758

123
456
7_8

123
456
78_

Sharada Koundinya,1BM23CS310
```