Lab 4

1. Write a Program to Implement 8-Puzzle problem using Python.

```
import heapq

# Goal state to compare
GOAL_STATE = [[1, 2, 3],
              [4, 5, 6],
              [7, 8, 0]]
```

```
# Movements: (dx, dy)
MOVES = {
    'UP': (-1, 0),
    'DOWN': (1, 0),
    'LEFT': (0, -1),
    'RIGHT': (0, 1)
}
```

```
class PuzzleState:
    def __init__(self, board, path=[], cost=0):
        self.board = board
        self.path = path
        self.cost = cost
        self.heuristic = self.calculate_heuristic()
        self.total_cost = self.cost + self.heuristic
```

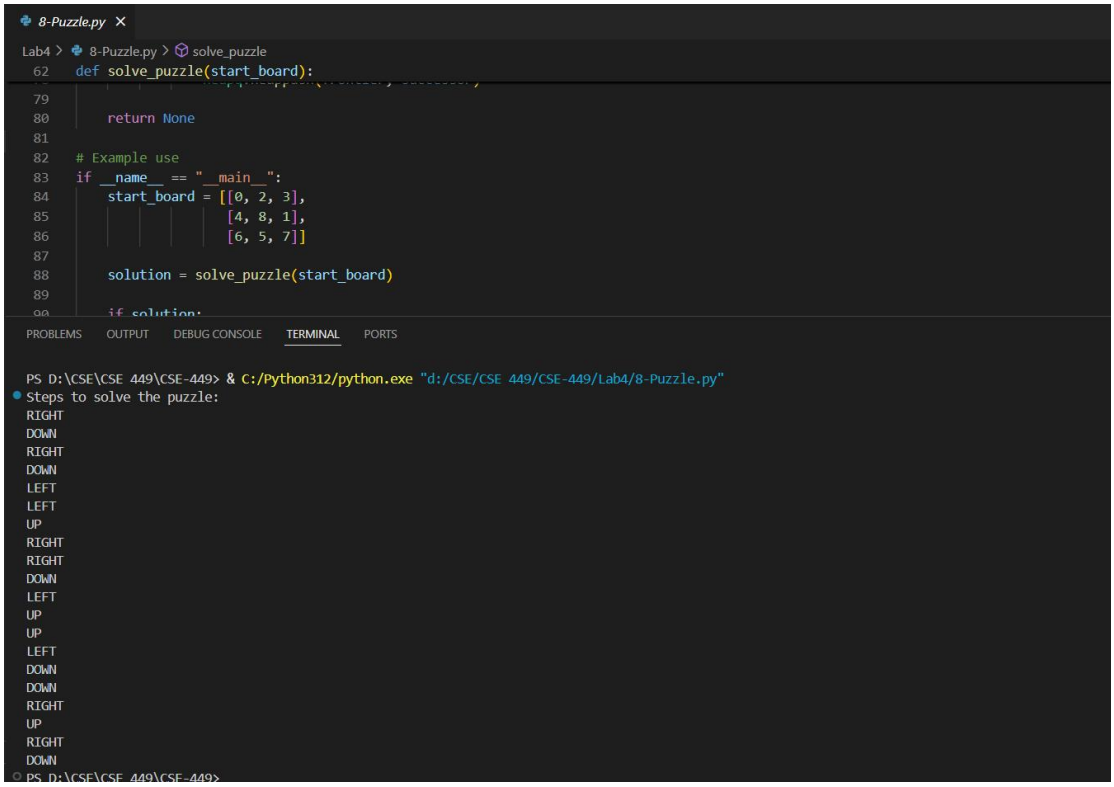
```
    def __lt__(self, other):
        return self.total_cost < other.total_cost
```

```
    def find_zero(self):
        for i in range(3):
            for j in range(3):
                if self.board[i][j] == 0:
                    return i, j
```

```
    def calculate_heuristic(self):
        # Manhattan Distance
        distance = 0
        for i in range(3):
            for j in range(3):
                value = self.board[i][j]
                if value != 0:
                    goal_x = (value - 1) // 3
                    goal_y = (value - 1) % 3
                    distance += abs(i - goal_x) + abs(j - goal_y)
        return distance
```

```
    def generate_successors(self):
        successors = []
        x, y = self.find_zero()
        for move, (dx, dy) in MOVES.items():
            new_x, new_y = x + dx, y + dy
            if 0 <= new_x < 3 and 0 <= new_y < 3:
                new_board = [row[:] for row in self.board]
                new_board[x][y], new_board[new_x][new_y] = new_board[new_x][new_y],
new_board[x][y]
                successors.append(PuzzleState(new_board, self.path + [move], self.cost + 1))
```

```
        return successors

    def is_goal(self):
        return self.board == GOAL_STATE

    def board_tuple(self):
        return tuple(tuple(row) for row in self.board)

def solve_puzzle(start_board):
    start_state = PuzzleState(start_board)
    frontier = []
    heapq.heappush(frontier, start_state)
    visited = set()

    while frontier:
        current_state = heapq.heappop(frontier)

        if current_state.is_goal():
            return current_state.path

        visited.add(current_state.board_tuple())

        for successor in current_state.generate_successors():
            if successor.board_tuple() not in visited:
                heapq.heappush(frontier, successor)

    return None

# Example use
if __name__ == "__main__":
    start_board = [[0, 2, 3],
                   [4, 8, 1],
                   [6, 5, 7]]

    solution = solve_puzzle(start_board)

    if solution:
        print("Steps to solve the puzzle:")
        for step in solution:
            print(step)
    else:
        print("No solution found.")
```



2. Write a Program to Implement Monkey Banana Problem using Python.

```
# Global Variable i
i = 0

def Monkey_go_box(x, y):
    global i
    i = i + 1
    print('step:', i, 'monkey slave', x, 'Go to ' + y)
```

```
def Monkey_move_box(x, y):
    global i
    i = i + 1
    print('step:', i, 'monkey take the box from', x, 'deliver to ' + y)
```

```
def Monkey_on_box():
    global i
    i = i + 1
    print('step:', i, 'Monkey climbs up the box')
```

```
def Monkey_get_banana():
    global i
    i = i + 1
    print('step:', i, 'Monkey picked a banana')
```

```
# Read the input operating parameters
codeIn = input(">> ")
codeInList = codeIn.split()
```

```
# The operating parameters indicate the locations of monkey, banana, and box respectively
monkey = codeInList[0]
banana = codeInList[1]
```

```
box = codeInList[2]
```

```
print('The steps are as follows:')
```

```
# Please use the least steps to complete the monkey picking banana task
Monkey_go_box(monkey, box)
Monkey_move_box(box, banana)
Monkey_on_box()
Monkey_get_banana()
```

```
PS D:\CSE\CSE 449\CSE-449> & C:/Python312/python.exe "d:/CSE/CSE 449/CSE-449/Lab4/MonkeyBanana.py"
>> door windows center
The steps are as follows:
step: 1 monkey slave door Go to center
step: 2 monkey take the box from center deliver to windows
step: 3 Monkey climbs up the box
step: 4 Monkey picked a banana
PS D:\CSE\CSE 449\CSE-449>
```