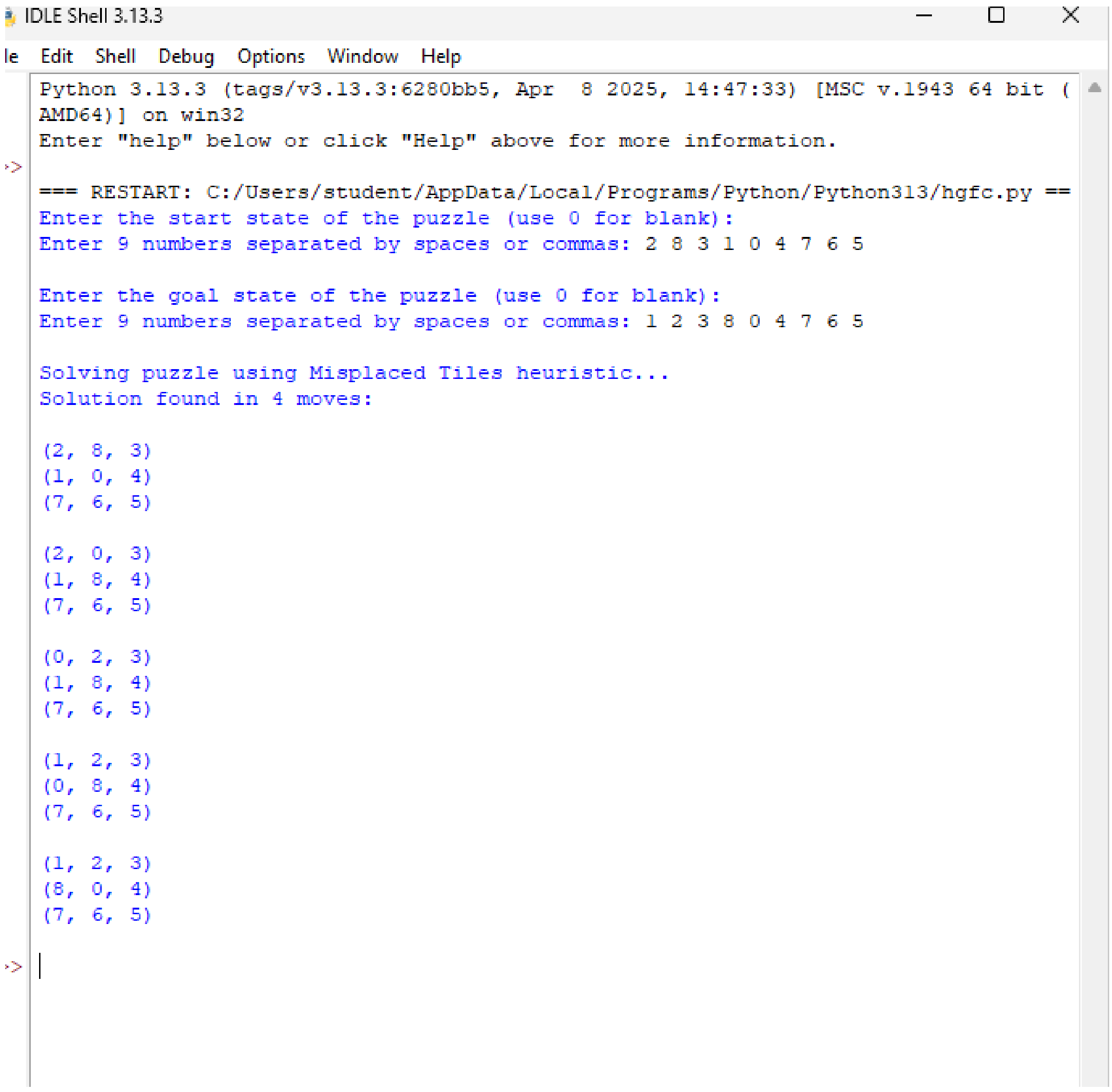
A* Using misplaced tiles

```
from heapq import heappush, heappop


def misplaced_tiles(state, goal_state):
    """Count tiles not in the correct position."""
    return sum(1 for i in range(9) if state[i] != 0 and state[i] != goal_state[i])


def get_neighbors(state):
    neighbors = []
    zero_pos = state.index(0)
    x, y = zero_pos // 3, zero_pos % 3
    moves = [(-1,0),(1,0),(0,-1),(0,1)]  # up, down, left, right

    for dx, dy in moves:
        nx, ny = x + dx, y + dy
        if 0 <= nx < 3 and 0 <= ny < 3:
            new_pos = nx * 3 + ny
            new_state = list(state)
            new_state[zero_pos], new_state[new_pos] = new_state[new_pos], new_state[zero_pos]
            neighbors.append(tuple(new_state))
    return neighbors


def a_star(start_state, goal_state, heuristic):
    open_set = []
    heappush(open_set, (heuristic(start_state, goal_state), 0, start_state, []))
    closed_set = set()

    while open_set:
        f, g, current, path = heappop(open_set)
        if current == goal_state:
            return path + [current]

        if current in closed_set:
            continue

        closed_set.add(current)

        for neighbor in get_neighbors(current):
            if neighbor in closed_set:
                continue
            new_g = g + 1
            new_f = new_g + heuristic(neighbor, goal_state)
            heappush(open_set, (new_f, new_g, neighbor, path + [current]))
    return None

def get_input_state(prompt):
    while True:
        try:
            raw = input(prompt)
            # Split by spaces or commas
            tokens = raw.replace(',', ' ').split()
            if len(tokens) != 9:
                print("Please enter exactly 9 numbers (0-8).")
                continue
            nums = tuple(int(x) for x in tokens)
            if set(nums) != set(range(9)):
                print("Numbers must be 0 through 8 with no duplicates.")
                continue
            return nums
        except ValueError:
            print("Invalid input, please enter integers only.")

def print_state(state):
    for i in range(0, 9, 3):
        print(state[i:i+3])
    print()

if __name__ == "__main__":
    print("Enter the start state of the puzzle (use 0 for blank):")
    start_state = get_input_state("Enter 9 numbers separated by spaces or commas: ")
    print("\nEnter the goal state of the puzzle (use 0 for blank):")
    goal_state = get_input_state("Enter 9 numbers separated by spaces or commas: ")

    print("\nSolving puzzle using Misplaced Tiles heuristic...")
    solution = a_star(start_state, goal_state, misplaced_tiles)

    if solution:
        print(f"Solution found in {len(solution) - 1} moves:\n")
        for step in solution:
            print_state(step)
    else:
        print("No solution found.")
```



```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr  8 2025, 14:47:33) [MSC v.1943 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>
=== RESTART: C:/Users/student/AppData/Local/Programs/Python/Python313/hgfc.py ==
Enter the start state of the puzzle (use 0 for blank):
Enter 9 numbers separated by spaces or commas: 2 8 3 1 0 4 7 6 5

Enter the goal state of the puzzle (use 0 for blank):
Enter 9 numbers separated by spaces or commas: 1 2 3 8 0 4 7 6 5

Solving puzzle using Misplaced Tiles heuristic...
Solution found in 4 moves:

(2, 8, 3)
(1, 0, 4)
(7, 6, 5)

(2, 0, 3)
(1, 8, 4)
(7, 6, 5)

(0, 2, 3)
(1, 8, 4)
(7, 6, 5)

(1, 2, 3)
(0, 8, 4)
(7, 6, 5)

(1, 2, 3)
(8, 0, 4)
(7, 6, 5)

>
```