D24DCS155

Practical-3

**Aim:** Imagine you are a wizard tasked with moving a set of magical rings from one tower to another. You have three towers:

• Tower A (the source)

• Tower B (the auxiliary)

• Tower C (the destination)

There are n rings of different sizes on Tower A. Your goal is to move all the rings to Tower C using the following rules:

1. You can only move one ring at a time.

2. You can only move the top ring of any tower.

3. No ring may be placed on top of a smaller ring.

To help you in this magical task, you will use a recursive algorithm. Implement it.

**Program (in Python):**

**def tower\_of\_hanoi(n, from\_rod, to\_rod, aux\_rod):**

**if n == 1:**

**print(f"Move disk 1 from {from\_rod} to {to\_rod}")**

**return**

**tower\_of\_hanoi(n-1, from\_rod, aux\_rod, to\_rod)**

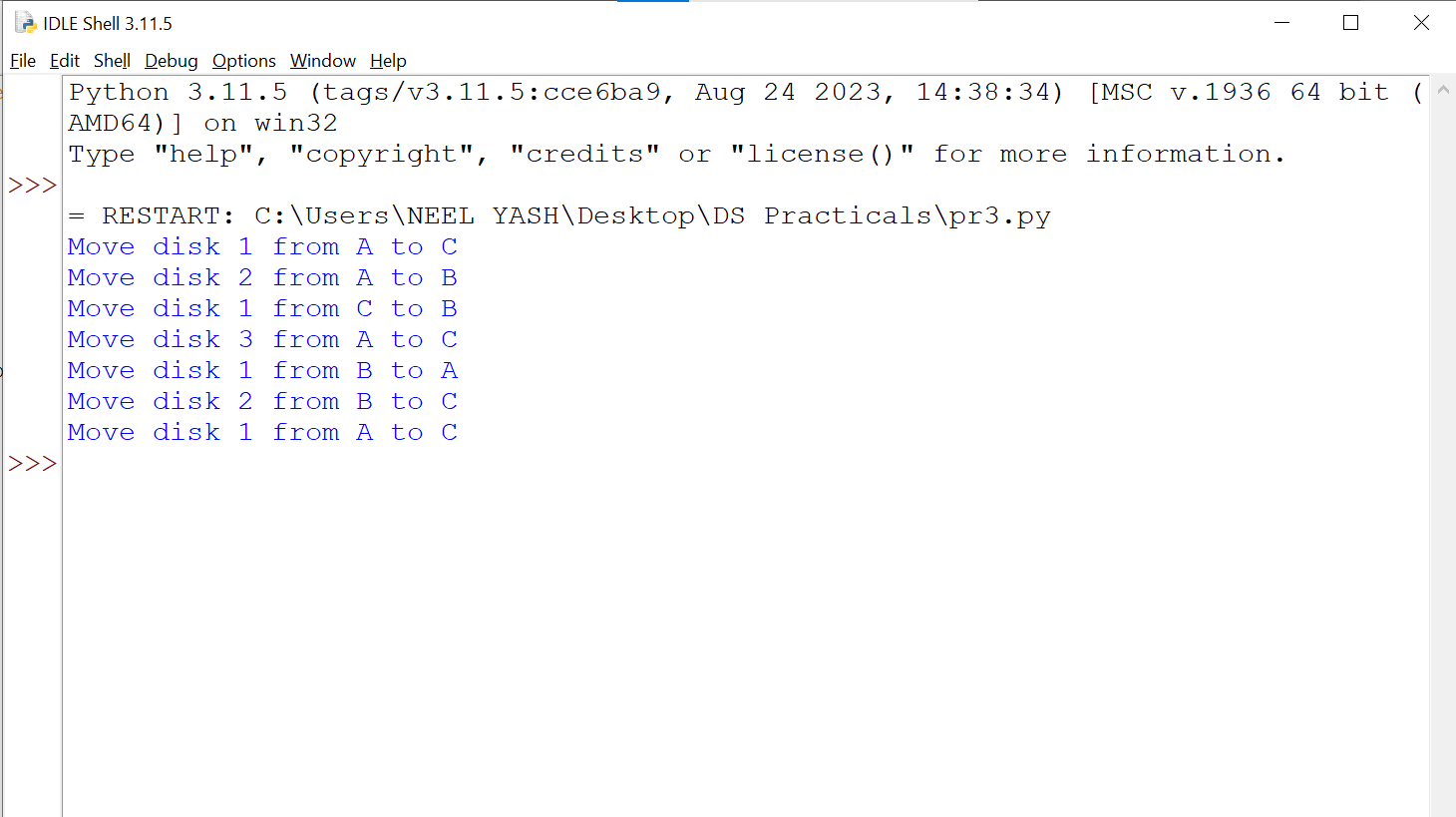
**print(f"Move disk {n} from {from\_rod} to {to\_rod}")**

**tower\_of\_hanoi(n-1, aux\_rod, to\_rod, from\_rod)**

**n = 3**

**tower\_of\_hanoi(n, 'A', 'C', 'B')**

**Output:**

****

**Fig.1: Tower Of Hanoi**

**Conclusion:**

In conclusion, the Tower of Hanoi program successfully implements the recursive algorithm to solve the classic problem. The program efficiently solves the problem by breaking it down into smaller sub-problems, utilizing the recursive function to move the disks. This implementation demonstrates the power of recursion in solving complex problems with a simple and elegant approach.