**TOWERS OF HANOI**

import sys

step = 0

def TowerOfHanoi(n, from\_rod, to\_rod, aux\_rod):

if n == 1:

move\_disk(from\_rod, to\_rod)

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

return

TowerOfHanoi(n-1, from\_rod, aux\_rod, to\_rod)

move\_disk(from\_rod, to\_rod)

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

TowerOfHanoi(n-1, aux\_rod, to\_rod, from\_rod)

def move\_disk(from\_rod, to\_rod):

disk = rods[from\_rod].pop()

rods[to\_rod].append(disk)

def print\_rods\_state():

for i in range(n, 0, -1):

row = []

for rod in ['A', 'B', 'C']:

if len(rods[rod]) >= i:

row.append(f"{rods[rod][i-1]} ")

else:

row.append("| ")

print("".join(row).rstrip())

print("A B C")

print()

def print\_step(action):

global step

step += 1

print(f"\n-->Step {step}: {action}")

print\_rods\_state()

def get\_positive\_integer():

while True:

try:

n = int(input("Enter the number of disks: "))

if n <= 0:

print("Please enter a positive integer.")

else:

return n

except ValueError:

print("Invalid input. Please enter a positive integer.")

sys.setrecursionlimit(1500)

n = get\_positive\_integer()

rods = {

'A': list(range(n, 0, -1)),

'B': [],

'C': []

}

print(f"\nTower of Hanoi Solution for {n} disks:")

print\_rods\_state()

# Start with the first move which will print the state after moving disk 1

TowerOfHanoi(n, 'A', 'C', 'B')

print(f"\nAll {n} disks have successfully moved from rod A to rod C")