Assignment On:
“ ADVANCED DATA STRUCTURES AND ALGORITHMS ”
(Assignment 4)

Submitted by:
Mohammed Nizamuddin
2503B05144
MTech CSE

Submitted to:
Dr. Rahul Mishra

Question:

1. **Solve the Towers of Hanoi Problem Recursively. Also, Implement a Stack-Based Iterative Solution.**

Code:

```
# ------------------------------------------------
# 1. RECURSIVE TOWERS OF HANOI
# ------------------------------------------------
def recursive_hanoi(n, source, auxiliary, destination):
    if n == 1:
        print(f"Move disk 1 from {source} → {destination}")
        return

    recursive_hanoi(n - 1, source, destination, auxiliary)
    print(f"Move disk {n} from {source} → {destination}")
    recursive_hanoi(n - 1, auxiliary, source, destination)


# ------------------------------------------------
# 2. ITERATIVE TOWERS OF HANOI USING STACKS
# ------------------------------------------------
def move_disk(from_rod, to_rod, from_name, to_name):
    disk = from_rod.pop()
    to_rod.append(disk)
```

```
    print(f"Move disk {disk} from {from_name} → {to_name}")


def legal_move(rod1, rod2, name1, name2):
    if not rod1:  # rod1 empty → move from rod2 to rod1
        move_disk(rod2, rod1, name2, name1)
    elif not rod2:  # rod2 empty → move from rod1 to rod2
        move_disk(rod1, rod2, name1, name2)
    elif rod1[-1] < rod2[-1]:  # smaller disk moves
        move_disk(rod1, rod2, name1, name2)
    else:
        move_disk(rod2, rod1, name2, name1)


def iterative_hanoi(n):
    # Stacks (rods)
    A = list(range(n, 0, -1))  # Start rod (largest → smallest)
    B = []
    C = []

    total_moves = (2 ** n) - 1

    print("\nIterative Solution:")

    # Swap auxiliary and destination for even number of disks
    if n % 2 == 0:
        aux_name, dest_name = "C", "B"
    else:
        aux_name, dest_name = "B", "C"

    for move in range(1, total_moves + 1):
        if move % 3 == 1:
            legal_move(A, C, "A", "C")
        elif move % 3 == 2:
            legal_move(A, B, "A", "B")
        else:
            legal_move(B, C, "B", "C")


# ------------------------------------------------
```

```
# MAIN PROGRAM (RUN BOTH SOLUTIONS)
# ------------------------------------------------
if __name__ == "__main__":
    n = int(input("Enter number of disks: "))

    print("\n--- Recursive Towers of Hanoi ---")
    recursive_hanoi(n, "A", "B", "C")

    print("\n--- Iterative Towers of Hanoi (Using Stacks) ---")
    iterative_hanoi(n)
```

Output:

```
Enter number of disks:  3

--- Recursive Towers of Hanoi ---
Move disk 1 from A → C
Move disk 2 from A → B
Move disk 1 from C → B
Move disk 3 from A → C
Move disk 1 from B → A
Move disk 2 from B → C
Move disk 1 from A → C

--- Iterative Towers of Hanoi (Using Stacks) ---

Iterative Solution:
Move disk 1 from A → C
Move disk 2 from A → B
Move disk 1 from C → B
Move disk 3 from A → C
Move disk 1 from B → A
Move disk 2 from B → C
Move disk 1 from A → C
```