Problem Statement:

Part-A (6 Marks):

The Towers of Hanoi is one of the most famous classic problems every budding computer scientist must grapple with. Legend has it that in a temple in the Far East, priests are attempting to move a stack of golden disks from one diamond peg to another (Fig. A). The initial stack has 64 disks threaded onto one peg and arranged from bottom to top by decreasing size. The priests are attempting to move the stack from one peg to another under the constraints that exactly one disk is moved at a time and at no time may a larger disk be placed above a smaller disk. Three pegs are provided, one being used for temporarily holding disks. Supposedly, the world will end when the priests complete their task, so there is little incentive for us to facilitate their efforts.
Let's assume that the priests are attempting to move the disks from peg 1 to peg 3. We wish to develop an algorithm that prints the precise sequence of peg-to-peg disk transfers. If we were to approach this problem with conventional methods, we would rapidly find ourselves hopelessly knotted up in managing the disks. Instead, attacking this problem with recursion in mind allows the steps to be simple. Moving n disks can be viewed in terms of moving only $n – 1$ disks (hence, the recursion), as follows:
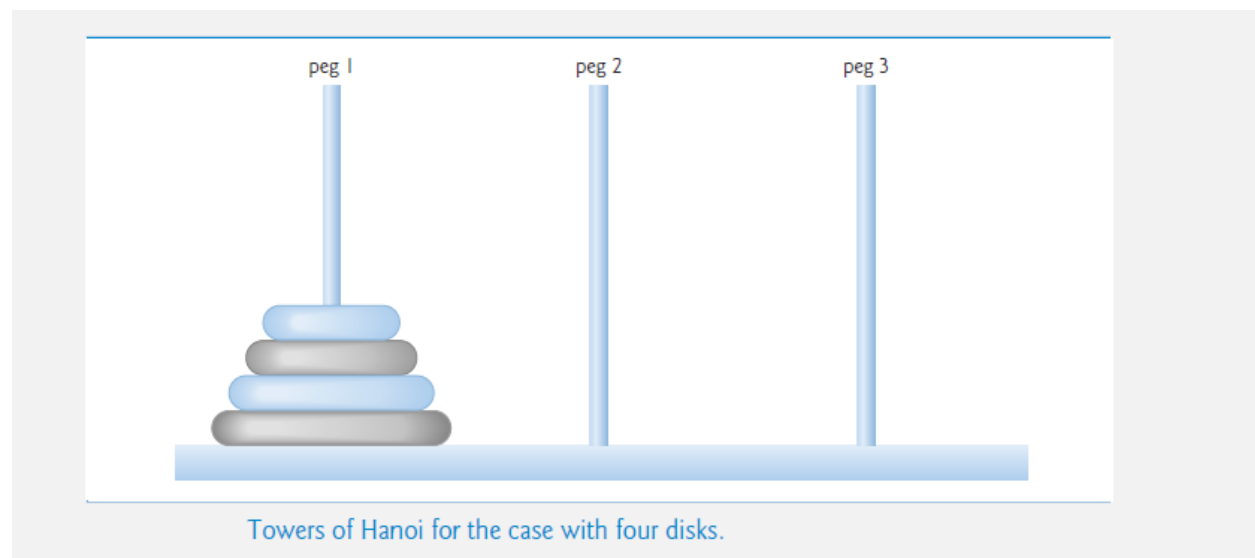

Towers of Hanoi for the case with four disks.

Figure. A

a) Move $n – 1$ disks from peg 1 to peg 2, using peg 3 as a temporary holding area.
b) Move the last disk (the largest) from peg 1 to peg 3.
c) Move the $n – 1$ disks from peg 2 to peg 3, using peg 1 as a temporary holding area.

The process ends when the last task involves moving $n = 1$ disk (i.e., the base case). This task is accomplished by simply moving the disk, without the need for a temporary holding area. Write a program to solve the Towers of Hanoi problem. Use a recursive function with four parameters:
a) The number of disks to be moved
b) The peg on which these disks are initially threaded
c) The peg to which this stack of disks is to be moved
d) The peg to be used as a temporary holding area

Display the precise instructions for moving the disks from the starting peg to the destination
peg. To move a stack of three disks from peg 1 to peg 3, the program displays the following moves:
1 → 3 (This means move one disk from peg 1 to peg 3.)
1 → 2
3 → 2
1 → 3
2 → 1
2 → 3
1 → 3

Part-B (3 Marks):

Any program that can be implemented recursively can be implemented iteratively, although sometimes
with more difficulty and less clarity. Try writing an iterative version of the Towers of Hanoi. If you
succeed, compare your iterative version with the recursive version developed in Part-A. Investigate
issues of performance, clarity and your ability to demonstrate the correctness of the programs.

Non-Coding-Part (1 Mark):

Create a **text** file named reflect.txt that contains your ***detailed*** description of the topics that
you have learned in completing this Lab and mention any issues that caused you difficulty and
how you solved them.