**Assignment 3: Control structures: recursion and iteration**

The power of computers and computing comes from the ability to execute the same task, or different versions of the same task, repeatedly. Programs and algorithms use iteration to perform repetitive jobs without requiring a large number of similar steps to be specified individually, such as “Do the next step 1000 times”. Closely related to repetition is recursion, a technique in which a concept is defined, directly or indirectly, in terms of itself.

**For this assignment, you will explore both iteration and recursion to solve the Tower of Hanoi** --a mathematical puzzle that was popularized by the French mathematician Edouard Lucas in 1883 and is consider a classic computer science problem.

**Tower of Hanoi**: The game consists of three poles (or “rods”) and a number of disks of different sizes which can slide onto any poles. The objective of the puzzle is to move the entire stack to another pole, obeying the following simple rules:

* You can only move one disk at a time.
* A disk cannot be put on top of a smaller disk.

The game begins with all of the disks stacked on a single pole, starting from the smallest to the biggest disk. The challenge lies in moving this arrangement to another pole. Only one disk can be moved at a time, and they all need to be moved to another pole, preserving the conical arrangement.



**Assignment overview:** For this assignment, you will design and code **two separate programs**: one which uses iteration and another which uses recursion to solve the Tower of Hanoi puzzle.

**Requirements**

1. Design, in pseudo code, Tower of Hanoi algorithms
2. In the programming language of your choice:
   1. Present a programmatic solution to the Tower of Hanoi using iteration
   2. Present a programmatic solution to the Tower of Hanoi using recursion
3. Provide a written report which:
   1. Introduces the assignment and the problem to be solved
   2. Briefly discusses the steps you took and decisions you made to start and solve the problem
   3. Discusses: did the pseudo code help you design the code? Briefly explain your answer. Which version of the program works “best”; discuss your answer.
   4. Concludes the report: wrap up the report (summary of assignment and your work)

**Deliverables:** You will submit a single Word or PDF document which includes:

* 1. The written discussion elements found in #3 (in the above section)
  2. Your pseudo code
  3. Video of each program/version (the working URL to the video should be listed in the report document; at the end of the report):
     1. compiling
     2. Showing the programs run from start to finish with the output of the solution for each version
  4. List (working URLs are sufficient) all resources used to complete this work (you are expected to explore at least two sources)
  5. Text of your code copied at the end of the report

**Expectations:**

* 1. **Compiling errors:** Your solutions must compile
  2. **Readability.** Your code should meet basic readability principles:
     1. Separate each component/part with white space.
     2. Align everything in a meaningful way.
  3. **Comments**: you must include enough comments to ensure that the code is described in sufficient detail such that anyone else looking at the code can easily understand the design and the purpose of the code.
  4. **All code and work associated with this assignment is your own work/written by you**.

.