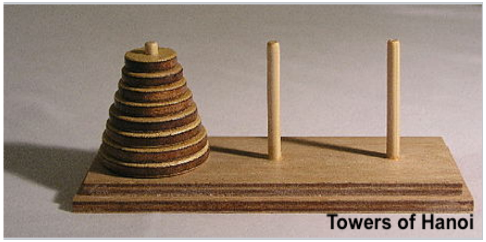
This quiz has 3 problems

Code 1) [9%]

The Towers of Hanoi is a mathematical puzzle whose solution illustrates recursion. There are three pegs which can hold stacks of disks of different diameters. A larger disk may never be stacked on top of a smaller. Starting with n disks on one peg, they must be moved to another peg one at a time. For n number of disks, the smallest number of steps (*hanoi*) to move the stack can be defined as the following function. For the below program, write a Fortran function (called **`ToH`**) to take an integer-number argument and return *hanoi.*

$$\mathrm{hanoi}(n) = \begin{cases} 1 & \text{if } n = 1 \\ 2 \cdot \mathrm{hanoi}(n-1) + 1 & \text{if } n > 1 \end{cases}$$



Code 2) [9%]

Use any iterative procedure (e.g. http://fourier.eng.hmc.edu/e176/lectures/ch2/node4.html) to determine the value of x at which the following function becomes zero:

$$6x^3 + 900x + 500$$

Your computed x has to be accurate to at least 8 significant digits and the absolute value of the function at your computed x has to be at most 1.e-6. Print x and the number of iterations on the screen.

Code 3) [12%]

Write a program that read an unknown size 1D array file (similar to attached data.txt), and process the following tasks for the first n elements of the array.

I. Identifies the size of the array.

II. Asks user to enter n following the below messages on screen.

```
Please enter the number of elements you want to process.
your entry should be an integer number >1 and < “size of the array”
```

III. Stops if the entry is out of the range.

IV. Print on the screen the sum of the first n elements and the minimum element value within this n elements.

V. From the first n elements, create a new array from only positive (>0) elements.

VI. Create a text file called `output.txt`, and write the new array made in step **V** on this file.

VII. Print on the screen the sum of the positive elements found in step **V** and the maximum element value.

Hint: intrinsic functions: `count`, `minval/maxval`, and `sum` could be helpful (https://gcc.gnu.org/onlinedocs/gfortran/)

Copy & paste your codes into the provided space on eClass Quiz#2-Nov09