Lab 4 :
Iterative vs recursive algorithms

Exercise 1 : Tower of Hanoi

"The tower of Hanoi" is a classic problem in computer science. It shows the power and readability of recursively defined algorithms. It is a puzzle game which consists of moving disks of different diameters from a tower of "start" to a "finish" tower via an "intermediate" tower and this in a minimum of moves[1], while respecting the following 2 rules:

a. We cannot move more than one disk at a time;
b. We can only place a disk on another disk larger than it or on an empty slot.

1. Write a recursive algorithm that solves the Towers of Hanoi problem. We assume that there are n disks to move (n is a natural number, n>=1).
2. Calculate the complexity of this algorithm.
3. Write the corresponding program in C language.
4. Measure running times T (in seconds) for a data sample of variable n and represent the results as a table :

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
T(s)																							

n	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	…	…	…	64
T(s)																			

5. Represent by a graph the variations of time T depending on the values of n.
6. Compare theoretical complexity and experimental measurements.
7. Are the theoretical predictions compatible with the experimental measurements?
8. Consider the iterative algorithm of the Towers of Hanoi problem. Write the corresponding program then calculate the running times (same table as question 4. to fill in) and represent the time variations by a graph.
9. Compare the recursive and iterative versions. What do you notice?

[1] For 64 disks we will have 2^{64}-1 moves. Assuming: 1 second to move a disk (86,400 movements per day). The end of the game would be after 213,000 billion days (584.5 billion years).

Exercise2 : Fibonacci Sequence

Consider the Fibonacci sequence $(U_n), n \in \mathbb{N}$ given by :

$$\begin{cases} U_0 = 0;\ U_1 = 1 \\ \forall n \geq 2: U_n = U_{n-1} + U_{n-2} \end{cases}$$

a. Write a *Fibo_Rec* function which recursively calculates the nth term of the Fibonacci sequence. Give its complexity.

b. Write a *Fibo_iter* function which iteratively calculates the nth term of the Fibonacci sequence of linear complexity.

c. Measure the running times T for the sample of numbers n below and complete the table :

n	5	10	15	20	25	35	45	60	100
T(n) of Fibo_Rec									
T(n) of Fibo_iter									

d. Compare the theoretical and experimental measurements of the two functions. Are the theoretical predictions compatible with the experimental measurements?

e. Represent with a graph the variations of the running time T(n) of *Fibo_Rec* and *Fibo_iter*. What do you notice?

f. We can show that the sequence $(U_n), \in \mathbb{N}$ satisfies $U_n \sim \phi^n$,
where ϕ is the golden ratio. So $\lim_{n\to+\infty} \frac{U_n}{U_{n-1}} = \phi$.

Make a copy of the *Fibo_iter* function and modify it to test this property (use the type double).

At each iteration, the function must display:

$$U_n \text{ and } \frac{U_n}{U_{n-1}}$$

Note : $\phi = \frac{1+\sqrt{5}}{2} \cong 1,6180339887498948482045868343656$