



Tutorial N° 08 : Recursion

Exercise 1 :

This exercise illustrates the importance of the order of a recursive call.

When $n=5$, what is displayed after executing the following two recursive procedures?

<pre> Procedure Display1 (n : integer); begin if (n>=1) then write (n) ; Display1(n-1) ; EndIf; EndProc; </pre>	<pre> Procedure Display2 (n : integer); begin if (n>=1) then Display2(n-1) ; write(n) ; EndIf; EndProc; </pre>
--	---

Exercise 2:

Write a recursive subalgorithm that calculates the sum of n first integers starting from 1.

Exercise 3:

Write a recursive subalgorithm which calculates the sum of two natural integers a and b (same for their product).

Exercise 4:

Write a recursive subalgorithm that calculates the sum of the digits of a natural number n .

Exercise 5:

Write a recursive subalgorithm that returns the i^{th} digit of a natural number n , starting from the least significant digit.

Exercise 6:

Write a recursive subalgorithm that calculates the quotient of the division of a by b (where a and b are two natural numbers).

Exercise 7:

Write a recursive subalgorithm that calculates the factorial of a natural number n .

Exercise 8:

Write a recursive subalgorithm that calculates the number of digits of a natural number n .

Additional Exercises

Exercise 9:

Write a recursive subalgorithm that calculates the average of n elements of an array.

Exercise 10 :

Ecrire un sous-algorithme récursif qui vérifie si un tableau de taille n est trié dans l'ordre croissant.

Exercise 11:

Write a recursive subalgorithm that checks if an array of size is sorted in ascending order.

Exercise 12:

Write a recursive subroutine that calculates the GCD (*Greatest Common Divisor*) of two strictly positive integers a and b.

Exercise 13:

Write a recursive sub-algorithm that calculates the Fibonacci sequence defined as follows:

$$\begin{cases} U_0 = 0 \\ U_1 = 1 \\ U_n = U_{n-1} + U_{n-2}, \forall n > 1 \end{cases}$$

Exercise 14

Write a recursive subalgorithm that calculates a to the power of b (a and b natural numbers).

Exercise 15:

Write a recursive subalgorithm that calculates the remainder of dividing a by b.

Exercise 16:

Write a directly recursive subalgorithm that checks whether a given integer is even or not.

Exercise 17:

Write an indirectly recursive subalgorithm which checks whether a given integer is even or not (i.e. a cross call between two recursive sub-algorithms: Even and Odd).

Exercise 18:

Write an algorithm which checks if a string is palindrome, by first giving the recursive subalgorithm which allows the mirror of this string to be calculated.

Exercise 19:

Write an algorithm that checks if a natural number is palindrome, by first giving the recursive subalgorithm that calculates the mirror of this number.

Exercise 20:

Write recursive subalgorithms that calculate the sum and product of the elements of an array of size n.