
TP 7 – Fonctions récursives

Présentation du problème

Une fonction récursive est une fonction qui s'appelle elle-même.

Tant que l'un des appels n'a pas renvoyé de résultat, l'exécution du programme garde en mémoire les appels en attente de résultat.

Cela autorise la réalisation d'algorithmes puissants pour réaliser certaines tâches.

Exercice 1 — Quatre façons de programmer la fonction factorielle

1. Programmer la fonction factorielle à l'aide d'une boucle séquentielle.
2. Programmer la fonction factorielle à l'aide d'une boucle conditionnelle.
3. Programmer la fonction factorielle à l'aide d'une fonction récursive.
4. Programmer la fonction factorielle à l'aide d'une fonction récursive avec accumulateur.

Dans une fonction récursive, un accumulateur sert à passer des informations supplémentaires permettant d'améliorer l'algorithme.

Exercice 2 — Trois façons de programmer la suite de Fibonacci

1. Programmer une fonction qui donne le terme de rang n de la suite de Fibonacci à l'aide d'une boucle séquentielle.
2. Programmer une fonction récursive qui réalise la même tâche.
3. Programmer une fonction récursive qui réalise la même tâche avec accumulateur.

Exercice 3 — PGCD par l'algorithme d'Euclide

La fonction ci-dessous permet de calculer le PGCD (*plus grand commun diviseur*) de deux entiers naturels.

```
1 def PGCD(a, b):  
    if b==0:  
3         return a  
    while b!=0:  
5         a, b = b, a%b  
    return a
```

Écrire une version récursive de cette fonction.