

Boucles. Arithmétique

Exercice 1 *Écrire la fonction factorielle de manière itérative.*

Exercice 2 *Écrire une fonction qui prend un entier en paramètre et qui retourne 1 si le nombre est premier, 0 sinon.*

Exercice 3 *Écrire une fonction qui calcule les coefficients binomiaux :*

$$C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Expliquer l'inconvénient majeur de cette formule, et transformer la fonction pour l'éviter.

Exercice 4 *Écrire une fonction `void decomposition(int n)` qui donne la décomposition de l'entier `n` en produit de facteurs premiers. Par exemple, l'exécution de `decomposition(60)` produira l'affichage :*

`60 = 2 * 2 * 3 * 5.`

Exercice 5 *Écrire une fonction `int EstSommeDeCarres(int n)` qui teste si l'entier `n` peut s'écrire comme somme de deux carrés d'entiers.*

Exercice 6 Calculs du ppcm et du pgcd

1. *Écrire une fonction qui utilise l'algorithme d'Euclide pour calculer le pgcd de deux entiers positifs de façon itérative.*
2. *Écrire une fonction `int ppcm(int a, int b)` qui calcule le plus petit multiple commun. On donnera une première version utilisant la fonction `pgcd` ; puis une seconde calculant directement le ppcm.*

Exercice 7 Problème de Syracuse

On définit, sur l'ensemble des entiers strictement supérieurs à 1, la fonction f suivante :

- *si n est pair, $f(n) = \frac{n}{2}$,*
- *sinon, $f(n) = 3n + 1$.*

On se donne un entier $u > 1$ et on considère la suite de ses itérés par la fonction f :

$$u, f(u), f(f(u)), f(f(f(u))), \dots$$

Par exemple, pour $u = 2$, on obtient : 2, 1 ; pour $u = 3$, on obtient : 3, 10, 5, 16, 8, 4, 2, 1 ; et pour $u = 7$, on obtient : 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1. On conjecture (mais cela n'a jamais été démontré) que quel que soit le nombre de départ, on aboutit à 1.

1. *Écrire une fonction itérative `void afficher(int u)` affichant les itérés successifs de l'entier u par la fonction f .*
2. *Écrire une fonction `int syracuse(int u)` retournant le nombre d'éléments de la suite des itérés de l'entier u par la fonction f . On écrira une version itérative et une version récursive de cette fonction.*