

TD 2 : Itérations et récursivité

Exercice 1 Factorielle

Concevoir et écrire en langage C, les versions itératives et récursives de la fonction factorielle qui renvoie la factorielle $n!$ d'un nombre n passé en paramètre. Rappel : $n! = 1 \times 2 \times \dots \times n$ si $n \geq 1$, et $0! = 1$.

Exercice 2 Fibonacci

Concevoir et écrire en langage C, les versions itératives et récursives de la suite de Fibonacci définie par : $U_0 = 0$, $U_1 = 1$, $U_{n+2} = U_n + U_{n+1}$

Exercice 3 PGCD

Le calcul du PGCD par l'algorithme d'Euclide est un algorithme fondamentalement récursif. En effet, il utilise la propriété suivante :

$$\text{pgcd}(a, b) = \text{pgcd}(b, a \bmod b)$$

Toute fonction récursive doit contenir une condition d'arrêt. Pour le PGCD, on a :

$$\text{pgcd}(a, 0) = |a|$$

Programmer la fonction PGCD de manière récursive et itérative.

Exercice 4 Pyramide

Spécifier et programmer une fonction `dessine_triangle` qui prend en argument un entier n et affiche un triangle isocèle comportant n lignes comme dans l'exemple suivant (où $n = 5$) :

```

      *
     ***
    *****
   ********
  *********
 
```

Exercice 5 Primalité

1. Spécifier et programmer une fonction qui renvoie un booléen indiquant si un entier a divise un autre entier n .

2. Spécifier et programmer une fonction qui renvoie un booléen indiquant si un entier est premier ou non.

NB: on rappelle qu'un nombre premier est un entier naturel, admettant exactement deux diviseurs distincts entiers et positifs : 1 et lui-même.

3. Programmer une fonction main qui affiche à l'écran tous les nombres premiers inférieurs à 100.