

Réversivité

M. Tellene

EXERCICE 1

Pour convertir un nombre entier positif n de la base décimale à la base binaire, il est possible d'opérer avec des divisions successives du nombre n par 2. Les restes des divisions constituent la représentation binaire.

Ecrire une fonction récursive `binaire(n)` permettant de calculer la représentation binaire d'un nombre n .

La fonction devra renvoyer une liste où chaque élément est un bit de la représentation binaire.

EXERCICE 2

La suite de Fibonacci est définie comme suit :

$$F_n = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F_{n-1} + F_{n-2} & \text{sinon} \end{cases}$$

Ecrire une fonction récursive `fib(n)` permettant de calculer l'élément n de la suite de Fibonacci.

EXERCICE 3

Décrire l'arbre d'appels de `fib(6)`

EXERCICE 4

Un nombre n est pair si $(n - 1)$ est impair, et un nombre n est impair si $(n - 1)$ est pair.

Ecrire deux fonctions récursives mutuelles `pair(n)` et `impair(n)` permettant de savoir si un nombre n est pair et si un nombre n est impair.

EXERCICE 5

Un nombre premier est un entier naturel qui admet exactement deux diviseurs distincts entiers et positifs : 1 et le nombre considéré, puisque tout nombre a pour diviseurs 1 et lui-même, les nombres premiers étant ceux qui ne possèdent pas d'autre diviseur.

Ecrire une fonction récursive `nb_premier(n, i)` renvoyant `True` si n est premier, `False` sinon. Il est à noter que la variable i sera incrémentée à chaque appel afin de tester la divisibilité de n .

EXERCICE 6

Le PGCD de a et b est le plus grand nombre qui est un diviseur à la fois de a et de b .

Ecrire une fonction récursive `pgcd(a, b)` renvoyant le PGCD de deux entiers a et b .