

**Exercice 1** Convertir en base 2 un nombre donné en base 10

L'algorithme de conversion par la méthode des divisions successives est le suivant :

$a$  est le nombre à convertir (un entier naturel)

$q$  est la variable qui contiendra les quotients successifs

Au départ :  $q = a$ .

$r$  est la variable qui contiendra les restes successifs

On appelle  $d$  la chaîne des chiffres binaires (le résultat).

Au départ,  $d$  est la chaîne vide ''.

Tant que  $q > 0$

    Faire la division euclidienne de  $a$  par 2.

    Concaténer le reste  $r$  à droite de  $d$ .

    Mettre le quotient  $q$  dans  $a$ .

Afficher  $d$

1) Compléter le programme de la fonction **dec2bin** suivante, grâce aux commentaires :

```
def dec2bin(a):
```

```
    nb = ..... # on mémorise la valeur du nombre à convertir
```

```
    d = ..... # d est la chaîne vide
```

```
    q = a
```

```
    while ..... : # on fait des divisions successives de a par 2 tant que q est non nul
```

```
        r = .....
```

```
        q = .....
```

```
        print(a, " = 2*", q, "+", r) # affiche a = 2*q + r
```

```
        d = str(r) + ..... # on convertit r est une chaîne puis on le colle à la chaîne d
```

```
        a = ..... # a prend la valeur du quotient q
```

```
    print("Le nombre ", nb, " s'écrit en base 2 : ", ..... ) # affichage du résultat
```

```
    return(d)
```

2) Tester la fonction précédente.

**Exercice 2** Convertir en base  $b$  un nombre donné en base 10.

1) En adaptant la fonction précédente, programmer une fonction **dec2base** (  $a$ ,  $b$  ) qui convertit un nombre entier naturel  $a$  donné en base  $b$  ( $2 \leq b \leq 9$ ).

2) Adapter la fonction précédente pour créer une fonction **dec2hexa** ( $a$ ) qui convertit un nombre entier naturel  $a$  donné en base 16. Cette fonction doit retourner une chaîne de caractères représentant le nombre en base 16, avec éventuellement des caractères A, B, C, D, E, F qui remplaceront respectivement les valeurs 10, 11, 12, 13, 14, 15.

### **Exercice 3** Convertir en base 10 un nombre donné en base 2.

#### **On utilise les puissances successives de 2.**

Le nombre écrit en base 2 est donné sous la forme d'une chaîne de caractères binaires de la forme '10010011' par exemple.

Les caractères d'une chaîne C sont numérotés de gauche à droite, à partir de 0.

On accède au caractère n° k par l'instruction C [ k ].

Si C = '10010011', C [ 3 ] renvoie '1'.

Pour obtenir la valeur décimale d'un nombre bin écrit en binaire, il faut parcourir la chaîne en partant de la droite, donc d'abord trouver le rang du bit le plus à droite (bit de poids faible).

Le parcours de la chaîne peut se faire avec une boucle for, le pas étant - 1.

Les puissances successives de 2 seront stockées dans la variable puiss.

Compléter le programme de la fonction **bin2dec(bi)** suivante :

```
def bin2dec(bi):
    deb = ..... # deb contient le rang du bit le plus à droite
    puiss = 1 # puiss contient initialement 1 = 20.
    n = 0
    for i in range(deb,.....,.....):
        n = n + int(bi[....])*puiss
        puiss = puiss*..... # calcul de la puissance de 2 suivante
    return(n)
```

### **Exercice 4** Convertir en base 10 un nombre donné en base b.

1) En adaptant le programme précédent, créer une fonction base2dec( nb, b ) qui convertit en base 10 le nombre nb, écrit en base b ( $2 \leq b \leq 9$ ) sous forme d'une chaîne.

2) En adaptant le programme précédent, créer une fonction hexa2dec( nb ) qui convertit en base 10 le nombre nb, écrit en base 16 sous forme d'une chaîne.