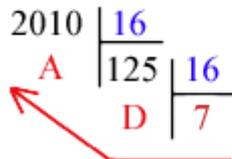


Exercice 1 :**De décimal en hexadécimal**

Le principe est le même que le schéma du cours, mais en divisant par 16 au lieu de 2.



Ainsi 2010 en base 10 s'écrit 7DA en base 16 (Rappelons que A=10 et D=13).

D'hexadécimal en décimal

16^3	16^2	16^1	16^0
A	4	0	F

Le nombre en base 10 est $10 \cdot 16^3 + 4 \cdot 16^2 + 15 = 40960 + 1024 + 15 = 41999$.

Exercice 3 :

Permet de convertir un nombre X de la base Y dans la base Z. ($2 \leq Y \leq 16$ et $2 \leq Z \leq 16$)

```
def convertirNombre(nbr,bd,ba):
    "nbr = nombre a convertir , bd = la base de depart, ba = la base d'arrivee"
    symboles = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F']
    nb10 = 0    # Nombre en base dix

    # Passage de la base de depart (bd) a la base 10.
    longueur=len(nbr)
    for i in range(longueur):
        nb10 = nb10 + symboles.index(nbr[i])*bd**(longueur-i-1)
    # print("(nombre en base 10 : ",nb10,")")

    # Passage de la base 10 a la base d'arrivee (ba).
    nombre = []
    if nb10<ba :
        resultat = nb10
    else :
        while nb10 >= ba:
            nombre.append(symboles[nb10 % ba])
            nb10 = nb10 // ba
            if nb10 < ba:
                nombre.append(symboles[nb10])
        nombre.reverse()
        resultat = ''
        for i in nombre:
            resultat = resultat + str(i)

    print("Le nombre",nbr,"en base",bd,"converti en base",ba,"donne",resultat)

convertirNombre('E3',16,15)
```

Permet de convertir un nombre X de la base Y dans la base Z. ($2 \leq Y \leq 16$ et $2 \leq Z \leq 16$)

```
def convertirNombre(nbr,bd,ba):
    "nbr = nombre à convertir , bd = la base de depart, ba = la base d'arrivée"
    symboles = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F']
    nb10 = 0 # Nombre en base dix

    # Passage de la base de depart (bd) à la base 10.
    longueur=len(nbr)
    for i in range(longueur):
        nb10 = nb10 + symboles.index(nbr[i])*bd**(longueur-i-1)
    # print("(nombre en base 10 :",nb10,")")

    # Passage de la base 10 à la base d'arrivee (ba).
    nombre = []
    if nb10<ba :
        resultat = nb10
    else :
        while nb10 >= ba:
            nombre.append(symboles[nb10 % ba])
            nb10 = nb10 // ba
        if nb10 < ba:
            nombre.append(symboles[nb10])
        nombre.reverse()
        resultat = ""
        for i in nombre:
            resultat = resultat + str(i)

    print("Le nombre",nbr,"en base",bd,"converti en base",ba,"donne",resultat)

convertirNombre('11011',2,16)
```

Exercice 3 :

```
456 se code sur 16 bits : 00000001 11001000
-1 se code sur 8 bits : 11111111
-56 se code sur 8 bits : 11001000
-5642 se code sur 16 bits : 11101001 11110110

01101100 = 108
11101101 = -19
10101010 10101010 = -21846
```

Exercice 4 :

- Le nombre de secondes est limité à $2^{31} - 1$, soit 2'147'483'647 secondes (01111111 11111111 11111111 11111111 en binaire).
- Ce nombre maximum sera atteint le 19 janvier 2038 à 3 h 14 min 7 s.
- Dans la seconde suivante, la représentation du temps « bouclera » (10000000 00000000 00000000 00000000 en binaire) et représentera $-2'147'483'648$ en complément à deux, et ainsi l'ordinateur affichera la date du 13 décembre 1901.

Exercice 5 :

Transformez $0,0101010101_2$ en base 10 :

0,333007812

Transformez $11100,10001_2$ en base 10 :

28,53125

Exercice 6 :

Transformez $0,5625_{10}$ en base 2 :

0,1001

Transformez $0,15_{10}$ en base 2 :

0,00100110011001...

Transformez $12,9_{10}$ en base 2 :

1100,1110011001100...

style par défaut
