

Autotest n°1 – correction

EXERCICE 1 : Convertir en base 10 les nombres suivants, en montrant la technique utilisée.

- 1) 1100101_2 2) 1001110_2 3) 1312_4 4) $A9_{16}$

Solution :

64	32	16	8	4	2	1
1	1	0	0	1	0	1
1	0	0	1	1	1	0

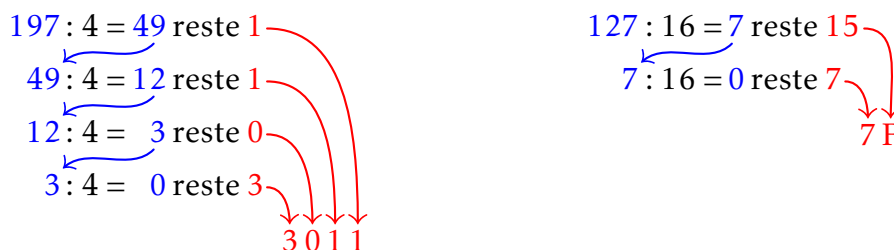
- 1) $1100101_2 = 64 + 32 + 4 + 1 = 101_{10}$
 2) $1001110_2 = 64 + 8 + 4 + 2 = 78_{10}$
 3) $1312_4 = 1 \times 64 + 3 \times 16 + 1 \times 4 + 2 \times 1 = 64 + 48 + 4 + 2 = 118_{10}$
 4) $A9_{16} = 10 \times 16 + 9 = 169_{10}$

EXERCICE 2 : Convertir dans la base demandée les nombres suivants, en montrant la technique utilisée.

- 1) 200 en base 2 2) 179 en base 2 3) 197 en base 4 4) 127 en base 16

Solution :

- 1) $200 = 128 + 72 = 128 + 64 + 8 = 11001000_2$
 2) $179 = 128 + 51 = 128 + 32 + 19 = 128 + 32 + 16 + 2 + 1 = 10110011_2$



EXERCICE 3 : Établir les tables de vérité des expressions booléennes suivantes :

- 1) $((\text{non } a) \text{ ou } b) \text{ et } (a \text{ ou } (\text{non } b))$ 2) $a \text{ ou non}((\text{non } a) \text{ ou } (\text{non } b))$

Solution :

a	b	$\text{non } a$	$\text{non } b$	$(\text{non } a) \text{ ou } b$	$a \text{ ou } (\text{non } b)$	$((\text{non } a) \text{ ou } b) \text{ et } (a \text{ ou } (\text{non } b))$
0	0	1	1	1	1	1
0	1	1	0	1	0	0
1	0	0	1	0	1	0
1	1	0	0	1	1	1

a	b	$\text{non } a$	$\text{non } b$	$(\text{non } a) \text{ ou } (\text{non } b)$	$\text{non}((\text{non } a) \text{ ou } (\text{non } b))$	$a \text{ ou non}((\text{non } a) \text{ ou } (\text{non } b))$
0	0	1	1	1	0	0
0	1	1	0	1	0	0
1	0	0	1	1	0	1
1	1	0	0	0	1	1

EXERCICE 4 : Un entier est représenté en hexadécimal avec 3 chiffres. Sans connaître sa valeur, combien de bits faut-il au minimum pour être sûr de pouvoir représenter ce nombre en binaire ?

Solution : Le plus grand nombre qu'on peut écrire avec 3 chiffres en hexadécimal est FFF_{16} .

Or $FFF_{16} = 15 \times 16^2 + 15 \times 16 + 15 = 4095_{10}$.

En binaire, $4095_{10} = 111111111111_2$.

Il faut donc 12 bits pour être sûr de pouvoir écrire le nombre.

EXERCICE 5 : On suppose qu'on a déjà défini une fonction `prochain(x)` qui prend un entier `x` et retourne un entier à l'aide d'un processus de calcul non précisé, définissant ainsi une suite de nombres. On considère la fonction `maximum(depart, n)` qui retourne la plus grande valeur de la suite parmi les `n+1` premières valeurs de la suite, en partant de `depart`, qui peut être le maximum.

```
1 def maximum(depart, n):
2     x = depart
3     val_max = x
4     for i in range(n):
5         x = prochain(x)
6         if x > val_max:
7             val_max = x
8     return val_max
```

Compléter les lignes 3, 6 et 7 pour que la fonction soit correcte.

EXERCICE 6 : On considère la fonction ci-dessous.

```
def mystere(n):
    i = 0
    s = 0
    while s < n:
        i = i + 1
        s = s + i
    return s
```

i	s	s < n
0	0	True
1	1	True
2	3	True
3	6	True
4	10	True
5	15	True
6	21	False

- 1) Compléter le tableau ci-contre pour l'exécution de `mystere(20)` et entourer la valeur renvoyée à la fin. Chaque ligne, à part la première, correspond à l'état de la mémoire après l'instruction `s = s + i`
- 2) Quelle est la valeur renvoyée par `mystere(100)` ?

Solution : On remarque que `s` est la somme des entiers de 1 à `i`. Lorsque `i = 14`, on obtient `s = 105` qui est donc la valeur obtenue.

EXERCICE 7 : Pour chacune des questions suivantes, déterminer la bonne réponse.

- 1) Choisir une expression booléenne pour la variable `S` qui satisfait la table de vérité suivante.

a) `A ou (non B)`

b) `(non A) ou B`

c) `(non A) ou (non B)`

d) `non (A ou B)`

A	B	S
0	0	1
0	1	0
1	0	1
1	1	1

- 2) Après l'exécution de ces affectations, que vaut `n` ?

a) 0

b) 1

c) 1.4

d) 2.7

```
>>> n = 25
>>> n += 2
>>> n = n//5
>>> n = n%2
```

- 3) Après l'exécution de ces affectations, que vaut `val` ?

a) 127

b) "12304"

c) "01234"

d) "41230"

```
>>> val = "123"
>>> val = val + "0"
>>> val = "4" + val
```

4) Qu'est-ce qui est affiché si on appelle truc(42)?

- a) Yo
- b) Ya
- c) Yo puis Ya
- d) Ya puis Yo

```
def truc(n):  
    if n >= 42:  
        print("Yo")  
        n = 41  
    else:  
        print("Ya")  
        n = 42
```

5) Qu'est-ce qui est affiché si on appelle mention(13)?

- a) Passable
- b) Assez bien
- c) Echec
- d) Passable puis Assez bien

```
def mention(n):  
    if n >= 10:  
        print("Passable")  
    elif n >= 12:  
        print("Assez bien")  
    else:  
        print("Echec")
```

EXERCICE 8 : On veut écrire une fonction `remplace(texte, s1, s2)` qui renvoie une copie de `texte` où le symbole `s1` a été remplacé par le symbole `s2`.

```
>>> remplace('bonjour', 'o', 'X')  
'bXnjXur'
```

```
>>> remplace('bonjour', 'M', 'i')  
'bonjour'
```

1) Quel est le résultat de `remplace("radiateur", "a", "o")`?

Solution : On obtient "rodiateur".

2) Quel est le résultat de `remplace("radiateur", "o", "a")`?

Solution : On obtient "radiateur", car il n'y a pas de "o".

3) Compléter le code de la fonction ci-dessous :

```
def remplace(texte, s1, s2):  
    res = ""  
    for lettre in texte:  
        if lettre == s1:  
            res = res + s2  
        else:  
            res = res + lettre  
    return res
```