II - PLANCHE 1 — Écriture binaire

1 **1.** Donner l'écriture binaire et hexadécimale des entiers naturels inférieurs à 16.

2. a. Donner l'écriture décimale des nombres écrits en base 2 suivants : $\overline{1011}^2$, $\overline{1001}^2$, $\overline{101}^2$.

b. Donner l'écriture binaire des nombres 12, 42, 33, 108 (écrits en base 10).

c. Convertir en écriture décimale les nombres dont l'écriture hexadécimale est $\overline{12}^{16}$, $\overline{BE}^{16}$, $\overline{3AE}^{16}$ et $\overline{FFF}^{16}$.

d. Convertir en hexadécimal les nombres exprimés en base 2 suivants : $\overline{11001010}^2$ et $\overline{11011101}^2$.

3. On peut parfois trouver dans le commerce des montres dont l'affichage est un peu… particulier.

a. Quelle est la valeur maximale que peut afficher la ligne du haut ? La ligne du bas ?

b. Quelle heure est-il ?

c. Combien de lumières au maximum sont allumées sur chacune des lignes ? Préciser l'heure correspondante.

2 **1. a.** Donner la représentation binaire de 57 et 198.

b. Additionner les représentations binaires de 57 et 198, puis convertir le résultat en représentation décimale.

2. Ajouter $\overline{1011}^2$ à $\overline{101}^2$. Convertir le résultat en représentation décimale.

3. Multiplier $\overline{1010}^2$ par $\overline{11}^2$. Convertir le résultat en représentation décimale.

3 **1. a.** Donner l'écriture décimale des couples de nombres binaires suivants :

i. $(\overline{1011}^2, \overline{101}^2)$ **ii.** $(\overline{1100}^2, \overline{110}^2)$ **iii.** $(\overline{1111}^2, \overline{111}^2)$

b. On note $n = \overline{b_3b_2b_1b_0}^2$ et $m = \overline{b_3b_2b_1}^2$. Exprimer m en fonction de n.

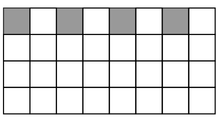
2. a. Donner l'écriture décimale des couples de nombres binaires suivants :

i. $(\overline{1011}^2, \overline{10110}^2)$ **ii.** $(\overline{1100}^2, \overline{11000}^2)$ **iii.** $(\overline{1111}^2, \overline{11110}^2)$

b. On note $n = \overline{b_3b_2b_1b_0}^2$ et $m = \overline{b_3b_2b_1b_00}^2$. Exprimer m en fonction de n.

3. On donne $n = \overline{1010}^2$. Donner l'écriture binaire de $m = 2n + 1$.

4 L'objectif de cet exercice est de remplir la grille ci-dessous.



Pour remplir les cases de l'image, vous devez utiliser la valeur binaire de la réponse à la question correspond à la ligne. Lorsque le bit est à 1 alors la case est grisée, lorsque le bit est à 0 alors la case est blanche.

1. Remplir la grille avec les informations.

a. Traduire la ligne 1 en binaire.

b. Convertir $\overline{24}^{10}$ en binaire.

c. Convertir $\overline{66}^{16}$ en binaire.

d. Convertir $\overline{3C}^{16}$ en binaire.

2. a. Faire un dessin dans une grille 8×8.

b. Donner 8 nombres en base 10 qui permettant de représenter votre dessin.

5 On donne le code de la fonction `mystere` ci-dessous.

Code python

```python
def mystere (a):
    k = 0
    while a > 0:
        if a % 10 == 7:
            k = k + 1
        a = a // 10
    return k
```

1. Que renvoie `mystere(75713)` ?

Justifier votre réponse à l'aide d'une trace d'exécution.

2. Écrire la spécification complète de la fonction `mystere`.

3. Écrire un jeu de 3 tests pour la fonction `mystere`.

6 **1.** On considère le code suivant :

Code python

```python
def bin_vers_deci(binaire):
    """ ... -> ...
    précondition : binaire est un nombre positif constitué de 0 et de 1 uniquement
    Renvoie le nombre binaire écrit en base 10 """
    expo = 0
    deci = 0
    while binaire > 0:
        dernier_chiffre = (binaire % 10)
        deci = deci + dernier_chiffre * ...
        binaire = ...
        expo = ...
    return ...
```

a. Recopier et compléter le code.

b. Écrire un jeu de trois tests pour la fonction `bin_vers_dec`. Tester la fonction.

2. On considère le code suivant :

Code python

```python
def deci_vers_bin(decimal):
    """ ... -> int
    précondition : décimal est un entier positif ou nul
    Renvoie le nombre décimal écrit en base 2
    """
    binaire = ""
    while decimal > 0:
        if ...:
            binaire = '1' + binaire
        else:
            ...
        decimal = ...
    return int(binaire) # on convertit la chaîne de caractère en entier
```

a. Recopier et compléter le code.

b. Écrire un jeu de trois tests pour la fonction `dec_vers_bin`. Tester la fonction.

c. Dresser la trace d'exécution de l'appel `dec_vers_bin(23)`.

7 Les lois de Morgan (Augustus De Morgan (1806-1871), mathématicien britannique) sont les propriétés suivantes :
Soient a et b deux booléens.

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

Démontrer la première loi de Morgan à l'aide des tables suivantes :

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

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

8 On appelle "xor" le "ou exclusif". On donne ci-dessous sa table de vérité :

a	b	a xor b
0	0	0
0	1	1
1	0	1
1	1	0

Ainsi, pour deux booléens a et b l'expression a xor b s'évalue vaut 1 lorsque a ou b valent 1, mais pas en même temps.

1. Exprimer a xor b uniquement à l'aide des opérations et, ou, et non.

2. En python, si `a` et `b` sont deux booléens, alors `a ^ b` est le xor de `a` et de `b`.

Déterminer en quoi s'évaluent les expressions suivantes.

a. `False ^ True`
b. `False ^ False`
c. `True ^ a` si `a = False`
d. `True ^ a` si `a = True`
e. `False ^ a`
f. `a ^ a`

3. Si `n` et `m` sont deux entiers, alors `n ^ m` correspond à l'entier dont l'écriture en base 2 est obtenue en effectuant le xor bit à bit sur les écritures en base 2 des entiers `n` et `m`.

a. Compléter le tableau ci-dessous.

`n` en base 2	1	0	1	0	1	0
`m` en base 2	1	1	0	1	0	1
xor bit à bit						

b. Quelle est l'écriture en base 10 de `n` ? De `m` ?
c. En déduire `n ^ m`.

4. Calculer `15 ^ 9`, `14 ^ 10`.