

Devoir surveillé n°1 – correction

EXERCICE 1 : (2pt) En utilisant le tableau ci-contre, déterminer la valeur en base 10 des nombres binaires donnés.

128	64	32	16	8	4	2	1	Décimal
0	0	1	1	0	0	1	0	50
1	0	0	0	0	1	0	1	133
1	0	0	0	1	0	1	0	138
0	0	0	0	1	1	1	1	15

EXERCICE 2 : (2pt) En utilisant le tableau ci-contre, déterminer l'écriture binaire des entiers donnés.

Décimal	128	64	32	16	8	4	2	1
13	0	0	0	0	1	1	0	1
46	0	0	1	0	1	1	1	0
95	0	1	0	1	1	1	1	1
162	1	0	1	0	0	0	1	0

EXERCICE 3 : (1,5pt) Convertir en base 10 les nombres suivants.

- 1) 3301_4 2) 254_6 3) BF_{16}

Solution :

- 1) $3301_4 = 3 \times 64 + 3 \times 16 + 0 \times 4 + 1 \times 1 = 192 + 48 + 1 = 241_{10}$
 2) $254_6 = 2 \times 36 + 5 \times 6 + 4 \times 1 = 106_{10}$
 3) $BF_{16} = 11 \times 16 + 15 = 191_{10}$

EXERCICE 4 : (1,5pt) Convertir dans la base demandée les nombres suivants.

- 1) 135 en base 4 2) 113 en base 6 3) 171 en base 16

Solution :

EXERCICE 5 : (2pt) Compléter les tables de vérité des expressions booléennes suivantes :

a	b	non a	(non a) et b	(non a) ou ((non a) et b)
0	0	1	0	1
0	1	1	1	1
1	0	0	0	0
1	1	0	0	0

a	b	non a	non b	(non a) et b	a et (non b)	((non a) et b) ou (a et (non b))
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

EXERCICE 6 : (1,5pt) Chacune des questions suivantes est indépendante. Les réponses doivent être justifiées brièvement.

1) Quel est le plus grand entier qui peut être représenté en binaire avec 4 bits?

Solution : Le plus grand nombre pouvant être écrit avec 4 bits est $1111_2 = 15_{10}$.

2) Combien de bits au minimum faut-il pour représenter en binaire 147?

Solution : Puisque $128 < 147 < 256$, il faut au moins 8 bits pour écrire 147.

3) Combien de lignes faut-il dans la table de vérité d'une expression booléenne avec 3 booléens, comme "a ou b ou c"?

Solution : Il y a 8 combinaisons possibles, donc 8 lignes.

EXERCICE 7 : (1,5pt) 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 `minimum(depart, n)` qui retourne la plus petite valeur de la suite parmi les `n+1` premières valeurs de la suite, en partant de `depart`, qui peut être le minimum.

```

1 def minimum(depart, n):
2     x = depart
3     val_min = x
4     for i in range(n):
5         x = prochain(x)
6         if x < val_min:
7             val_min = x
8     return val_min

```

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

EXERCICE 8 : (3pt) On considère la fonction ci-dessous.

```

def mystere(n):
    u = 1
    while u < n:
        u = 2*u + 1
    return u

```

1) Compléter le tableau ci-contre pour l'exécution de `mystere(50)` et entourer la valeur renvoyée à la fin.

2) Quelle est la valeur renvoyée par `mystere(1000)`? On pourra comparer les valeurs successives de `u` avec des puissances de 2.

Solution : On remarque que `u` est toujours égale à une puissance de 2, moins 1. La plus petite puissance de 2 supérieure à 1000 est 1024. Le résultat est donc 1023.

u	u < n
1	True
3	True
7	True
15	True
31	True
63	False

EXERCICE 9 : (3pt) Chez le dentiste, la bouche grande ouverte, lorsqu'on essaie de parler, il ne reste que les voyelles. Nous allons écrire une fonction `dentiste(texte)` qui renvoie un texte ne contenant que les voyelles de `texte`, dans l'ordre. Les voyelles sont a, e, i, o et u. Même les ponctuations sont supprimées.

```
>>> dentiste("j'ai mal")
'aia'
```

1) Quel est le résultat obtenu pour `dentiste("il fait chaud")`?

Solution : On obtient "iaiau".

2) Quel est le résultat obtenu pour `dentiste("")`?

Solution : On obtient "".

3) Pour écrire la fonction `dentiste(lettre)`, on pourra s'inspirer de la fonction `recopier(texte)` qui prend un texte et renvoie une copie de ce texte. On part du texte vide et on parcourt toutes les lettres de `texte` en les rajoutant à droite de `resultat`.

```
def recopier(texte):
    resultat = ""
    for lettre in texte:
        resultat = resultat + lettre
    return resultat
```

On dispose d'une fonction `voy(lettre)` qui renvoie un booléen indiquant si `lettre` est une voyelle ou non.

```
>>> voy("a")
True
>>> voy("m")
False
```

Compléter la fonction `dentiste(texte)` ci-dessous, en utilisant `voy(lettre)` :

```
def dentiste(texte):
    resultat = ""
    for lettre in texte:
        if voy(lettre):
            resultat = resultat + lettre
    return resultat
```

EXERCICE 10 : (3pt) Pour chacune des questions suivantes, déterminer la bonne réponse.

1) Soient P et Q deux formules logiques telles que P est vraie et Q est fausse.

Quelle est la valeur de l'expression (P et Q) ou ((non P) ou Q)?

- a) vraie
- b) **fausse**
- c) ni vraie, ni fausse
- d) vraie et fausse en même temps

2) Quelles valeurs des booléens *a*, *b* et *c* permettent de rendre vraie l'expression suivante :
((non *a*) et *b*) et ((non *b*) ou *c*)

- a) *a* = 1, *b* = 1 et *c* = 1
- b) *a* = 1, *b* = 0 et *c* = 0
- c) ***a* = 0, *b* = 1 et *c* = 1**
- d) *a* = 0, *b* = 0 et *c* = 0

3) Après l'exécution de ces affectations, que vaut *n*?

- a) 0
- b) 5
- c) **8**
- d) 8.125

```
>>> n = 5
>>> k = 8
>>> p = n + k
>>> n = n * p
>>> n = n // k
```

4) Après l'exécution de ces affectations, que vaut *val*?

- a) 250
- b) "40123"
- c) **"012340123"**
- d) "1234123"

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

5) Qu'est-ce qui est affiché si on appelle `truc(41)`?

- 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
```

6) Qu'est-ce qui est affiché si on appelle `mention(11)`?

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

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