

DEVOIR COMMUN N°2

L'UTILISATION DE LA CALCULATRICE EST INTERDITE

1 Manipulation de dictionnaires /3

Soit le dictionnaire défini par l'instruction :

```
>>> d = {'nom': 'Dupuis', 'prenom': 'Jacque', 'age': 30}
```

- 1) Un dictionnaire met en relation une clé et une valeur. Donner un exemple de clé et la valeur correspondante dans le dictionnaire ci-dessus.

0,5 ▶ Clé : 'nom', valeur : 'Dupuis'

- 2) On s'est trompé en renseignant le prénom, on voulait écrire **Jacques**. Écrire le code de l'instruction Python qui permet de corriger le prénom.

0,5 ▶

```
>>> d['prenom'] = 'Jacques'
```

- 3) Écrire le code de l'instruction Python qui permet d'écrire, *en utilisant le dictionnaire d* :

Jacques Dupuis a 30 ans.

0,5 ▶

```
>>> print("{} {} a {} ans.".format(d['prenom'], d['nom'], d['age']))
```

- 4) Écrire le code Python qui permet d'afficher la liste des clés du dictionnaire. *Ce code doit comporter une boucle et utiliser le dictionnaire d.*

0,5 ▶

```
>>> for cle in d.keys():  
    print(cle)
```

ou

```
>>> for cle in d:  
    print(cle)
```

- 5) Écrire le code Python qui permet de réaliser l'affichage ci-dessous. *Ce code doit comporter une boucle et utiliser le dictionnaire d.*

1
nom : Dupuis
prenom : Jacques
age : 30

▶

```
>>> for cle in d:
```

```
print("{} : {}".format(cle, d[cle]))
```

2 Représentation des nombres entiers /7

Partie 1. Dans cette partie on décide de coder les nombres entiers relatifs sur 10 bits *en utilisant la méthode du complément à 2*.

6) Combien de valeurs est-il possible de représenter ?

0,5 ▶ $2^{10} = 1024$

7) Quelle est la valeur la plus grande représentable ?

0,5 ▶ $2^{10-1} - 1 = 511$

8) Quelle est la valeur la plus petite représentable ?

0,5 ▶ $-2^{10-1} = -512$

9) Donner l'écriture binaire du nombre 63.

0,5 ▶ $63 = (0000111111)_2$

10) Donner l'écriture binaire du nombre -1 .

1 ▶ $-1 = (1111111111)_2$

11) Quel entier relatif a pour représentation binaire $(1000000000)_2$?

0,5 ▶ $(1000000000)_2 = -512$

12) Quel entier relatif a pour représentation binaire $(0110101010)_2$?

0,5 ▶ $(0110101010)_2 = 426$

13) Quelle est la représentation binaire de l'entier relatif opposé de l'entier relatif dont la représentation binaire est $(1000000001)_2$? Justifier la réponse.

1 ▶ $(0111111111)_2$
Méthode la plus rapide : on prend le complément à 1 de la représentation binaire proposée et on ajoute 1.
Autre méthode : $(1000000001)_2 = -511$ et $511 = (0111111111)_2$.

Partie 2. Dans cette partie tous les nombres sont des entiers naturels.

14) Quel nombre a pour représentation hexadécimale $(38)_{16}$?

0,5 ► $(38)_{16} = 56$

15) Quelle est la représentation hexadécimale du nombre dont la représentation binaire est $(11111010)_2$?

0,5 ► $(11111010)_2 = (FA)_{16}$

16) Quelle est la valeur du nombre obtenu à l'issue de l'opération $(111010)_2 + (011000)_2$?

1 ► $(111010)_2 + (011000)_2 = (1010010)_2 = 82$

3 Changement de base /6

On considère le code Python suivante :

```
def f(x, b):  
    s = 0  
    k = len(x) - 1  
    for a in x:  
        s = s + int(a) * b ** k  
        k = k - 1  
    return s
```

17) Quelle est la valeur initiale de la variable `k` lors de l'appel `f('0101', 2)` ? Justifier la réponse.

1 ► `k` vaut initialement 3.

18) Que retourne l'appel `f('0101', 2)` ? Justifier la réponse en indiquant les différentes valeurs prises par les variables `s`, `k` et `a`.

2 ►

s	k	a
0	3	'0'
0	2	'1'
4	2	'0'
4	1	'1'
5	0	

19) La définition de la fonction précédente fait intervenir une boucle `for`. Ré-écrire cette définition en remplaçant la boucle `for` par une boucle `while`.

►

```
def f(x, b):  
    s = 0  
    k = len(x) - 1  
    i = 0
```

1

```

while i < len(x):
    a = x[i]
    s = s + int(a) * b ** k
    k = k - 1
    i = i + 1
return s

```

20) Écrire le corps de la fonction `decimal_vers_binaire` dont la spécification est :

```

def decimal_vers_binaire(n):
    """ int -> str
    Retourne la chaîne de caractère représentation du nombre entier n dans la base 2.
    Les valeurs de poids le plus élevé sont placées en premier.

    >>> decompose(13)
    '1101'
    >>> decompose(0)
    '0'
    """

```

Tout élément de réponse sera pris en compte.

►

```

def decimal_vers_binaire(n):
    """ int -> str
    Retourne la chaîne de caractère représentation du nombre n dans la base 2.
    Les valeurs de poids le plus élevé sont placées en premier.

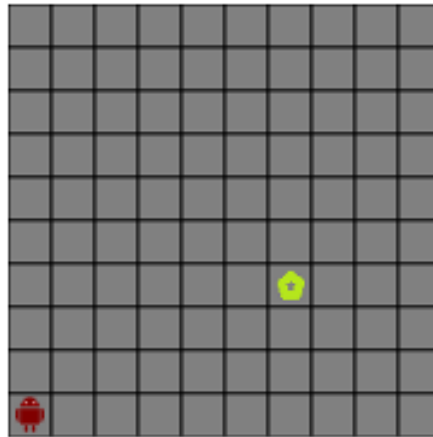
    >>> decompose(13)
    '1101'
    >>> decompose(0)
    '0'
    """
    res = ""
    if n == 0:
        res = '0'
    else:
        while n != 0:
            # ajout du reste de la division euclidienne au début de la liste
            res = str(n % 2) + res
            # poursuite la décomposition avec le quotient de la division
            n = n // 2
    return res

```

4 Pilotage d'un robot sur un quadrillage /4

On peut piloter un robot sur le quadrillage 10×10 ci-dessous grâce à quatre commandes : `haut()`, `bas()`, `droite()` et `gauche()`. Chacune de ces commandes déplace le robot d'une seule case dans la direction indiquée par le nom de la commande.

Le robot comprend une commande supplémentaire, `ramasser()`, lui permettant de ramasser la gem sur la grille.



- 21) Écrire un programme comportant des boucles `for`, qui commande au robot de *parcourir toutes les cases de la grille* et de ramasser la gem quand il passera dans la case la contenant.

Tout élément de réponse sera pris en compte.

4

```

► for j in range(1, 11): # permet de se déplacer vers le haut (10 lignes)
    for i in range(1, 10): # permet de se déplacer vers la droite
        if j % 2 != 0: # pour les rangs impairs
            droite()    # on se déplace d'une case vers la droite
        else:          # pour les rangs pairs
            gauche()    # on se déplace d'une case vers la gauche

        if j == 4 and i == 3: # sur la 4ème ligne et après 3 déplacements vers la
gauche
            ramasser()

    haut() # après 9 déplacements horizontaux on "monte" d'une case

```