

[le Barème donné est indicatif]

Exercice 1 (5 pt) : LISTES

On définit à titre d'exemple la liste suivante `l = ['a', 'b', 'c', 'b', 'a']`

Q1-1 (0.5 pt) : Qu'affiche ce code python 3 ? Combien de lignes seront affichées ?

```
for e in l:
    print(e)
```

Q1-2 (1.0 pt) : Donnez le code python 3 de la fonction `palindrome(l)` qui retourne `True` si la liste passée en paramètre peut se lire comme un palindrome, et `False` autrement.

Exemple : `palindrome(l)` pour la liste donnée en exemple retourne `True`.

Q1-3 (1.5 pt) : Donnez le code python 3 de la procédure `rotate_right(l)` qui modifie la liste `l` en sa rotation par décalage à droite d'une case (sans perte d'éléments).

Exemple : si `l=[1,2,3,4,5]` avant `rotate_right(l)`, alors, après appel `l=[5,1,2,3,4]`

Q1-4 (2.0 pt) : Donnez le code python 3 de la fonction `rotate_right(l,shift)` qui modifie la liste `l` par sa rotation par décalage à droite de `shift` cases, en réalisant un minimum d'opérations.

Exemple : si `l=[1,2,3,4,5]` avant `rotate_right(l,2)`, alors, après appel `l=[4,5,1,2, 3]`

Exercice 2 (5 pt) : TRANCHES DE LISTES et RANGES

On définit la liste suivante `l = ['a', 'b', 'c', 'd', 'e']`

Q2-1 (1.5 pt) : Quel est le type, et qu'affiche cette suite d'instructions dans l'interpréteur python 3 ?

```
>>>l[2:-2]
>>>l[-1:2:-1]
>>>l[:2] + ['bc'] + l[2:]
```

Q2-2 (1.0 pt) : Qu'affiche cette suite d'instructions dans l'interpréteur python 3 ?

```
>>>l[1] = ['bb'] ; l
>>>l[:2] = [] ; l
```

Q2-3 (1.0 pt) : Qu'affiche cette suite de 4 blocs d'instructions dans l'interpréteur python ?

```
>>> for i in range(5):
>>>     print(i,end=' ')
>>> for i in range(1,5):
>>>     print(i,end=' ')
>>> for i in range(1,5,-1):
>>>     print(i,end=' ')
>>> for i in range(5,1,-1):
>>>     print(i,end=' ')
>>>
```

Q2-4 (2.5 pt) : Donnez le code python 3 pour tester si un entier `e` est à la fois :

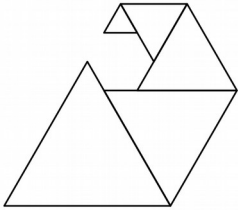
- compris entre 0 et 31 (inclus),
- est pair, mais n'est pas multiple de 4

(a) avec une seule expression booléenne courte (composition au choix : les « % », « and », « not », « <= », « >= », « == » ou tests bit à bit « & » et « | » vu en codage sont autorisés).

(b) à l'aide de deux tests d'appartenance à deux *ranges* (que vous définirez auparavant), et une expression booléenne utilisant seulement un « and » et un « not »

Exercice 3 (4 pt) : TURTLE niveau collèe

Voici un exercice du brevet des collèges de cette année

<pre>def triangles(): l = 100 for i in range(5): triangle(l) turtle.forward(l) l = l - 20</pre>	<pre>def triangle(l): turtle.pendown() for i in range(3): turtle.forward(l) turtle.left(120) turtle.penup()</pre>	
---	---	---

Q3-1 (1 pt): Donnez la longueur du deuxième et dernier triangles tracés par **triangles()**

Q3-2 (1 pt): Tracez à main levée l'allure de la figure obtenue quand on exécute **triangles()**

Q3-3 (2 pt): On modifie le script initial pour obtenir la *figure de droite*. Indiquez, dans **triangles()**, la ligne **après laquelle** on peut placer l'instruction **turtle.left(60)** pour obtenir cette nouvelle figure, en justifiant votre choix.

Exercice 4 (6 pt) : ANNUAIRE

On donne une très longue liste normalisée **ANNUAIRE = [{'nom': 'DUBOIS', 'prenom': 'KEVIN', 'tel': '0712345678'}, {'nom': 'DUPONT', 'prenom': 'PAUL', 'tel': '0387654321'}, ...]**

Q4-1 (0.5 pt) : Donnez le code (python 3) qui retourne le **nom** de la 1ere personne de cette liste.

Q4-2 (1.5 pt) :

- **(a)** Donnez un code calculant « le » ou « les » (si ex-aequo) **prénom(s)** le(s) plus fréquent(s) de la liste en indiquant le choix de la structure retournée.
- **(b)** Comment modifier votre code pour retourner la liste de tous les prénoms trouvés par fréquence décroissante d'utilisation ?

Q4-3 (3 pt) :

- **(a)** Donnez le code python3 permettant de savoir, pour un *prénom donné*, s'il apparait plus fréquemment avec un numéro de téléphone débutant par 07 que par 06.
- **(b)** Comment pré-calculer rapidement ces statistiques, pour ensuite avoir la réponse (instantanément) pour tout prénom demandé ?