Feuille d'exercices : Listes et tuples.

Exercice 1

Voici un script Python :

```
1 premiers = [2,3,5,7]
2 couple = (7,4)
3 i=3
4 a = premiers[i]
```

1. Quelle est la valeur de `premiers[2]` ? .....
2. Après l'exécution de `couple.append(1)`, quelle sera la valeur de `couple` ? .....
3. Après l'exécution de `premiers[3]=11`, quelle sera la valeur de `premiers` ? .....
4. Quelle instruction faut-il exécuter pour enlever la valeur 3 de `premiers` ? .....

Exercice 2

Voici un script Python :

```
1 lieux = ['Ville','Campagne','Foret']
2 couleurs = ['bleu','vert','rouge','marron']
3 liste  = ['0','0','1','1','1','2','0','1','2','3']
4 lettres = ['a','b','c','d']
5 triplet = ('pierre','feuille','ciseaux')
```

1. "Ville" est :

   ☐ Un index ☐ un item

2. La variable `lieux` est :

   ☐ une liste ☐ un tuple ☐ Un dictionnaire

3. Cocher les bonnes réponses :

   ☐ L'index de `couleur` est 5 ☐ L'index de `vert` est 2 ☐ L'index de `marron` est 3

4. `liste.count(1)` vaut :

   ☐ 0 ☐ 1 ☐ 4

5. `liste.count('2')` vaut :

   ☐ 0 ☐ 2 ☐ 3

6. Si on exécute `lettres.append('e')` alors `len(lettres)` vaut :

   ☐ 4 ☐ 5 ☐ 6

7. Cocher les opérations valides :

   ☐ `triplet.append(4)` ☐ `triplet[0]` ☐ `triplet[0] = 4`

Exercice 3

On donne le code de la fonction suivante :

```
1 def mystere(liste :list) -> bool :
2     """
3     :param liste d'éléments comparables
```

```
    """
    for i in range(1, len(liste)):
        # ICI
        if liste[i-1] > liste[i]:
            return False
    return True
```

1. Compléter le tableau ci-dessous à chaque passage par la ligne indiquée par le commentaire #ICI lors de l'appel de `mystere([1,2,6,9])`. Quel sera le retour de la fonction pour cet appel ?

i	i-1	liste[i-1]	liste[i]

2. Compléter le tableau ci-dessous à chaque passage par la ligne indiquée par le commentaire #ICI lors de l'appel de `mystere([1,6,2,9])`. Quel sera le retour de la fonction pour cet appel ?

i	i-1	liste[i-1]	liste[i]

3. On donne le code suivant :

```
a = mystere(['kiwi','pomme','ananas','framboise'])
b = mystere(['ananas','framboise','kiwi','pomme'])
```

Quel seront les valeurs des variables `a` et `b` ?

4. Expliquer ce que fait la fonction `mystere`.

Exercice 4

On considère le nombre

$$A_n = n^2 - 2n + 3 \qquad n \in \mathbb{N}$$

Écrire un programme qui stocke dans une liste `valeurs` les différentes valeurs de A_n pour n allant de 0 à 100.

Exercice 5

On considère un trinôme $P(x) = ax^2 + bx + c$ avec $a \neq 0$.
Coder une fonction nommée `racines` qui prend en paramètre un triplet `coef = (a,b,c)` et qui retourne un tuple dont le premier argument est le nombre de racines et où les autres argument sont les éventuelles racines de P.
On donne ci-dessous un code à compléter.

```
from math import sqrt
def racines(coef):
    a,b,c = coef
    delta = ...
    if delta <0 :
        resultat = ...
    else :
        if delta == 0:
            resultat = ...
        else : #delta >0
            racine1 = ...
            racine2 = ...
            resultat = ...
    return resultat
```

Compléter le code et écrire 3 tests permettant de vérifier le bon fonctionnement de cette fonction.

Exercice 6

1. Si liste désigne la liste `[1,[2,3],[4,5],6,7]`, que vaut `len(liste)` ? .....
2. Que vaut `[2*n for n in range(5)]` ? .....
3. Si `liste = [-5,2,3,-7,42,7]` , alors que vaut `[n for n in liste if n > 0]` ?

Parmi les codes suivants, lesquels permettent de construire la liste des cinq premiers nombres impairs ?

1.
```
impairs = [1,3,5,7,9]
```

2.
```
impairs = [2*n+1 for n in range(5)]
```

3.
```
impairs = [n for n in range(1,11,2)]
```

4.
```
impairs = []
for n in range(5):
    impairs.append(2*n+1)
```

5.
```
impairs = []
n= 0
while len(impairs) != 5:
    if n % 2 == 1 :
        impairs.append(n)
    n = n+1
```

Donner plusieurs programmes permettant de construire les 25 premiers nombres pairs

Exercice 8

Voici un code en Python :

```
lettres = ['a','b','c']
nombres = [1,5]
couples = [(c,n) for c in lettres for n in nombres]
```

Quelle est la valeur de `couples` à l'issue du script ?
Écrire un programme constitué de boucles qui permet d'obtenir la variable `couples`.

Exercice 9

On considère la matrice (tableau de nombres en mathématique) : $\begin{pmatrix} -3 & 1 & 0 \\ 1 & -1 & 2 \\ 0 & -3 & 5 \end{pmatrix}$

1. Créer une liste nommée a représentant la matrice A.
2. Écrire une fonction `valabs` qui prend en paramètre une liste et qui retourne une liste composé des items de la liste initiale sans leurs signes.
   On pourra utiliser la fonction abs() qui donne la valeur absolue d'un nombre donné en paramètre.
3. A l'aide de la fonction `valabs`, construire une liste b par compréhension représentant une matrice dont tous les coefficients sont égaux à ceux de A mais sans leurs signes.
4. Créer une fonction somme qui calcule la somme des nombres constituant une matrice passée en paramètre.

Exercice 10

Écrire une fonction `min_max_moy` admettant une liste comme argument et renvoyant la valeur minimale, la valeur maximale et la moyenne des valeurs de la liste. Le retour sera du type `tuple`.