

Listes et tuples - Plan de travail

1NSI – janvier 2023

Savoir-faire de la séquence

-

Ordre des étapes à respecter

1

1. Quel est le type de l'expression $f(4)$ si la fonction f est définie par

```
def f(x):  
    return (x, x**2)
```

- a) un entier b) un flottant c) une liste d) un tuple
2. On dispose d'une liste définie par $L = [15, 17, 12, 23]$.
Quelle est la valeur de L après l'instruction $L[2] = 25$

- a) $[15, 25, 12, 23]$ b) $[15, 17, 25, 12, 23]$
c) $[15, 25, 25, 23]$ d) $[15, 17, 12, 25, 23]$

3. Après l'affectation suivante :

```
alphabet = [ 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', ]
```

quelle est l'expression qui permet d'accéder à la lettre E?

- a) `alphabet.E` b) `alphabet['E']`
c) `alphabet[4]` d) `alphabet[5]`

4. On considère le script suivant :

```
t = [2, 8, 9, 2]  
t[2] = t[2] + 5
```

Quelle est la valeur de t à la fin de son exécution?

- a) $[2, 13, 9, 2]$ b) $[2, 8, 14, 2]$ c) $[2, 13, 14, 2]$ d) $[2, 13, 9, 2]$

5. On considère la fonction suivante :

```
def somme(tab):  
    s = 0  
    for i in range(len(tab)):  
        .....  
    return s
```

Par quelle instruction faut-il remplacer les points de suspension pour que l'appel `somme([10, 11, 12, 13, 14])` renvoie 60?

- a) `s = tab[i]` b) `s = s + tab[i]`
c) `tab[i] = tab[i] + s` d) `s = s + i`

6. On exécute le script suivant :

```
m = []  
for i in range(5):  
    n = []  
    for j in range(3):  
        n.append(i*j)  
        m.append(n)
```

Quelle est la valeur de m à la fin de son exécution?

- a) [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]
- b) [[0, 0, 0], [0, 1, 2], [0, 2, 4], [0, 3, 6], [0, 4, 8]]
- c) [[1, 1, 1], [2, 4, 6], [3, 6, 9], [4, 8, 12], [5, 10, 15]]
- d) [[1, 1, 1, 1, 1], [2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20], [5, 10, 15, 20, 25]]

7. L est une liste d'entiers. On définit la fonction suivante

```
def f(L):  
    m = L[0]  
    for x in L:  
        if x > m:  
            m = x  
    return m
```

Que calcule cette fonction?

- a) le maximum de la liste L passée en argument
- b) le minimum de la liste L passée en argument
- c) le premier terme de la liste L passée en argument
- d) le dernier terme de la liste L passée en argument

8. Quelle est la valeur de la variable r à la fin de l'exécution du script suivant?

```
t = (10,6,1,12,15)  
r = t[3] - t[1]
```

- a) -9
- b) 2
- c) 3
- d) 6

9. On définit : [10,9,8,7,6,5,4,3,2,1]

Quelle est la valeur de L[L[3]]

- a) 3
- b) 4
- c) 7
- d) 8

10. On considère la liste de listes suivantes

```
tictactoe = [['X', 'o', 'o'],  
             ['o', 'o', 'o'],  
             ['o', 'o', 'X']]
```

Quelle instruction permet d'obtenir une diagonale de "X"?

- a) tictactoe[3] = "X"
- b) tictactoe[4] = "X"
- c) tictactoe[1][1] = "X"
- d) tictactoe[2][2] = "X"

11. On définit ainsi une liste M

```
M = [['A', 'B', 'C', 'D'], ['E', 'F', 'G', 'H'], ['I', 'J', 'K', 'L']]
```

Quelle expression vaut la chaîne de caractères 'H'?

- a) M[1][3]
- b) M[3][1]
- c) M(7)
- d) M(8)

En mathématiques, la suite de Fibonacci est une suite d'entiers dans laquelle chaque terme est la somme des deux termes qui le précèdent.

Voici les premiers termes de la suite

1, 2, 3, 5, 8, 13, 21 ...

Écrire une fonction `Fibonacci` qui prend en argument un nombre n et qui retourne la liste des n premiers nombres de la suite de Fibonacci.

Une suite de Syracuse est une suite d'entiers naturels définie de la manière suivante : on part d'un nombre entier strictement positif ; s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et l'on ajoute 1. En répétant l'opération, on obtient une suite d'entiers strictement positifs dont chacun ne dépend que de son prédécesseur.

Par exemple, à partir de 14, on construit la suite des nombres :

14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2...

C'est la suite de Syracuse du nombre 14.

Après que le nombre 1 a été atteint, la suite des valeurs 1, 4, 2, 1, 4, 2... se répète indéfiniment en un cycle de longueur 3, appelé cycle trivial. On s'arrêtera donc au premier 1 rencontré.

1. Écrire une fonction `syracuse` qui prend en argument le nombre de départ puis qui renvoie la liste des termes de la suite de Syracuse associée à ce nombre. On s'arrêtera au premier 1 rencontré.
2. (a) Écrire une fonction `temps_de_vol` qui prend en argument la liste des termes de la liste et qui renvoie le nombre de termes avant d'arriver à 1.
(b) Lister les temps de vol de tous les nombres de 2 à 50.
3. (a) Écrire une fonction `altitude_de_vol` qui prend en argument la liste des termes de la liste et qui renvoie la plus haute valeur.
(b) Lister les altitudes de vol de tous les nombres de 2 à 50.
4. (a) Écrire une fonction `temps_en_altitude` qui prend en argument la liste des termes de la liste et qui renvoie le nombre de terme plus haut que le premier terme de la liste.
(b) Lister les temps en altitude de tous les nombres entre 2 et 50.