

**Contrôle de connaissances**

Nom : ..... Prénom : .....

Exercice 1 : ( 4 points )

On dispose du tableau suivant :

```
tab = [ 8, -5, 15, [9, 12, 17] , "abcd" , 3.7 , -24.2 , 18 ]
```

Que renvoie la console après les instructions suivantes :

(-1-) : `>>> tab[5]`

```
Solution :  
| 3.7
```

(-2-) : `>>> len(tab)`

```
Solution :  
| 8
```

(-3-) : `>>> tab[3][-1]`

```
Solution :  
| 17
```

(-4-) : `>>> 15 in tab[3]`

```
Solution :  
| False
```

(-5-) : `>>> tab[5:]`

```
Solution :  
| [3.7, -24.2, 18]
```

(-6-) : `>>> tab.append([8,1])`  
`>>> len(tab)`

```
Solution :  
| 9
```

(-7-) : `>>> for i in tab :`  
`... if type(i) == int and i < 0 :`  
`... print(i , end = "")`

 **Solution :**

| -5

(-8-) : &gt;&gt;&gt; tab.pop()

 **Solution :**

| [8, 1]

Exercice 2: ( 5 points )

On dispose de la fonction suivante suivant :

```
def une_fonction(liste) :  
    a = liste[0]  
    c = 1  
    for i in range(1, len(liste)) :  
        if liste[i] == a :  
            c += 1  
        if liste[i] > a :  
            a = liste[i]  
            c = 1  
  
    return c
```

Compléter le tableau suivant pour l'appel suivant :

```
>>> une_fonction([6,6,7,1,7,8,7,8])
```

Itérations	i	liste[i]	a	c
-	-	-	6	1
1	1			
2				
3				
4				
5				
6				
7				

**Correction**

Itérations	i	liste[i]	a	c
-	-	-	6	1
1	1	6	6	2
2	2	7	7	1
3	3	1	7	1
4	4	7	7	2
5	5	8	8	1
6	6	7	8	1
7	7	8	8	2

Expliquer, en justifiant, ce qui renvoie la fonction pour un tableau d'entiers quelconque.

### **Solution :**

La fonction renvoie le nombre de valeurs correspondant au maximum du tableau, car à chaque lecture de l'élément du tableau, si la valeur est identique au maximum mis en mémoire dans la variable `a`, le compteur `c` est incrémenté. Si cette valeur dépasse le maximum, ce maximum change, et le compteur reprend à 1.

Exercice 3 : ( 6 points )

On souhaite simuler une lancer de 100 dés de 6, en stockant les résultat dans une matrice  $10 \times 10$  :

Par exemple :

```
[[4, 3, 2, 1, 4, 5, 3, 6, 6, 6],
 [5, 3, 3, 3, 6, 3, 2, 3, 3, 3],
 [5, 5, 5, 2, 6, 1, 1, 5, 5, 1],
 [2, 4, 6, 4, 5, 1, 5, 5, 2, 1],
 [1, 4, 6, 5, 2, 5, 1, 4, 2, 4],
 [1, 1, 5, 1, 6, 2, 5, 5, 4, 6],
 [4, 2, 2, 4, 5, 5, 1, 5, 6, 1],
 [6, 2, 6, 6, 6, 6, 1, 3, 5, 3],
 [6, 3, 5, 2, 5, 6, 1, 6, 1, 1],
 [6, 3, 3, 4, 5, 5, 6, 3, 6, 6]]
```

Compléter la partie du code suivant pour obtenir une telle matrice :

```
from random import .....

t = [[..... (1,...) for i in range(.....)] for j in range(.....)]
```

**Correction**

```
from random import randint

t = [[randint(1,6) for i in range(10)] for j in range(10)]
```

On cherche maintenant à compter le nombre de fois où sur une **ligne** un 6 et encadré par 2 autres 6. ( ce qui en vrai dans l'exemple en position (0,8)...)

Compléter le programme suivant pour répondre à la question :

```
def fonction_ligne(tab) :
    compteur = .....
    for i in range(0 , len(tab)) :
        for j in range(1 , len(tab)-1) :
            if tab[...][.....] == 6 and \
                tab[i][j] ==6 and \
                tab[...][.....] == 6:
                    compteur = .....
    return .....
```

**Correction**

```
def fonction_ligne(tab) :
    compteur = 0
    for i in range(0 , len(tab)) :
        for j in range(1 , len(tab)-1) :
            if tab[i][j-1] == 6 and \
                tab[i][j] ==6 and \
```

```
        tab[i][j+1] == 6:  
            compteur = compteur + 1  
    return compteur
```

---

Exercice 4: ( 5 points)

Un gérant de parc d'attractions cherche à écrire une fonction pour la gestion des entrées. Il voudrait une fonction qui prend en argument le nombre d'adultes et le nombre d'enfants de moins de 10 ans. Il propose ces prix :

- 17 € par adultes.
- 13 € par enfants.

Il propose aussi les réductions suivantes :

- Si le nombre d'adultes est supérieur ou égal à 10, chaque adulte paye alors 15 €.
- Si le nombre d'enfants est supérieur ou égal à 10, chaque enfant paye alors 11 €.

Il a commencé par :

```
def prix_a_payer(nb_adultes , nb_enfants) :  
    prix = nb_adultes * 17 + nb_enfants * 13  
    return prix
```

Mais son programme ne prend pas en compte ses réductions.

Modifier la fonction `prix_a_payer` pour qu'elle réponde à la situation :

**Correction**Listing 1 – La fonction `prix_a_payer`

```
def prix_a_payer(nb_adultes , nb_enfants) :  
    if nb_adultes >= 10 :  
        prix_adulte = 15  
    else :  
        prix_adulte = 17  
    if nb_enfants >= 10 :  
        prix_enfant = 11  
    else :  
        prix_enfant = 13  
    prix = nb_adultes * prix_adulte + nb_enfants * prix_enfant  
    return prix
```