

Spécialité NSI Première	<b>DEVOIR SURVEILLE DE</b> <b>NSI</b> <b>N° 4</b>	Vendredi 16 décembre 2022
Lycée d'Avesnières		Durée : 55 mn
Année scolaire 2022-2023		Calculatrice interdite

**NOM** : .....

**Prénom** : .....

**Complétez directement sur l'énoncé** les programmes en langage Python.

## 1 Remplacer une valeur

Écrire une fonction `remplacer` prenant en argument :

- une liste d'entiers `valeurs`
- un entier `valeur_cible`
- un entier `nouvelle_valeur`

et renvoyant une **nouvelle** liste contenant les mêmes valeurs que `valeurs`, dans le même ordre, sauf `valeur_cible` qui a été remplacé par `nouvelle_valeur`.

**La liste passée en argument ne doit pas être modifiée.**

### Exemples

Si `valeurs = [3, 8, 7]`

alors `remplacer(valeurs, 3, 0)` renvoie `[0, 8, 7]` et `valeurs` vaut toujours `[3, 8, 7]`

Si `valeurs = [3, 8, 3, 5]`

alors `remplacer(valeurs, 3, 0)` renvoie `[0, 8, 0, 5]` et `valeurs` vaut toujours `[3, 8, 3, 5]`

```
def remplacer(valeurs, valeur_cible, nouvelle_valeur):
```

```
    return liste
```

## 2 Maximum

Écrire une fonction `maximum` prenant en argument :

- une liste non vide de nombres nombres

et renvoyant le plus grand élément de cette liste.

Chacun des nombres utilisés est de type `int` ou `float`. On interdit ici d'utiliser `max`, ainsi que `sort` ou `sorted`.

### *Exemple*

Si `nombres = [98, 12, 104, 23, 131, 9]`  
alors `maximum(nombres)` renvoie `131`

```
def maximum(nombres):
```

```
    return maxi
```

### 3 Indice de la première occurrence

Écrire une fonction `indice` prenant en argument :

- Un nombre entier nommé `element`
- Un tableau de nombres entiers nommé `tableau`

et renvoyant l'indice de la première occurrence du nombre `element` dans le tableau.

Si le nombre `element` est absent du tableau alors la fonction renvoie `None`.

On n'utilisera pas la fonction `index` ni la fonction `find`.

#### **Exemples**

Si `tableau = [10, 12, 1, 56]`

alors `indice(1, tableau)` renvoie `2`

Si `tableau = [1, 50, 1]`

alors `indice(1, tableau)` renvoie `0`

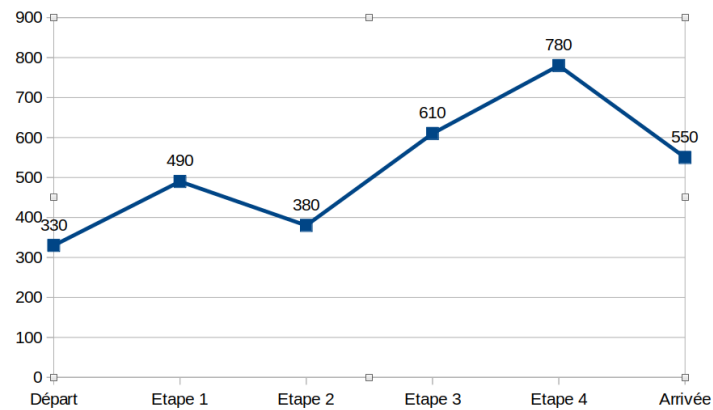
Si `tableau = [2, 3, 4]`

alors `indice(1, tableau)` renvoie `None`

```
def indice(element, tableau):
```

## 4 Dénivelé positif

Le dénivelé cumulé positif d'une course de montagne est la somme totale des dénivelés de l'ensemble des ascensions durant la course.



Écrire une fonction `denivele_positif` prenant en argument :

- Une liste `altitudes` non vide des altitudes atteintes à la fin de chaque ascension ou de chaque descente pendant la course

et renvoyant le nombre `cumul` positif qui est la somme de toutes les ascensions.

### Exemple

Si `altitudes = [330, 490, 380, 610, 780, 550]`

alors `denivele_positif(altitudes)` renvoie `560` En effet il y a trois ascensions :

la première de  $490 - 330 = 160$ , la deuxième de  $610 - 380 = 230$ , la troisième de  $780 - 610 = 170$  et le cumul vaut  $160 + 230 + 170 = 560$

```
def denivele_positif(altitudes):
```

```
    return cumul
```

## 5 Dernière occurrence

Écrire une fonction `derniere_occurrence` prenant en argument :

- Un tableau d'entiers non vide `tableau`
- Un entier `cible`

et renvoyant l'indice de la dernière occurrence de l'entier `cible` dans le tableau.

Si l'élément n'est pas dans le tableau alors la fonction renvoie la longueur du tableau.

### *Exemples*

Si `tableau = [2, 8, 5, 2, 4]`

alors `derniere_occurrence(tableau, 2)` renvoie `3`

Si `tableau = [5, 3, 7, 6]`

alors `derniere_occurrence(tableau, 2)` renvoie `4`

```
def derniere_occurrence(tableau, cible):
```

```
    return indice
```

## 6 Moyenne simple

Écrire une fonction `moyenne` prenant en argument :

- Un tableau d'entiers non vide `valeurs`

et renvoyant la moyenne des valeurs du tableau.

On n'utilisera pas la fonction `sum` ni de fonctions prédéfinies permettant de calculer directement la moyenne.

### *Exemples*

Si `valeurs = [1, 3]`

alors `moyenne(valeurs)` renvoie `2.0`

```
def moyenne(valeurs):
```

```
    return moyenne
```

## 7 Recherche des positions d'un élément dans un tableau

Écrire une fonction `indices` prenant en arguments :

- Un entier nommé `element`
- Un tableau d'entiers nommé `entiers`

et renvoyant la liste croissante des indices de l'entier `element` dans le tableau `entiers`.

Cette liste sera donc vide si `element` n'apparaît pas dans le tableau.

On n'utilisera pas les fonctions `index` et `max`.

### Exemples

Si `entiers = [3, 2, 1, 3, 2, 1]`

alors `indices(3, entiers)` renvoie `[0, 3]`

Si `entiers = [1, 1, 1, 1]`

alors `indices(1, entiers)` renvoie `[0, 1, 2, 3]`

Si `entiers = [1, 2, 3]`

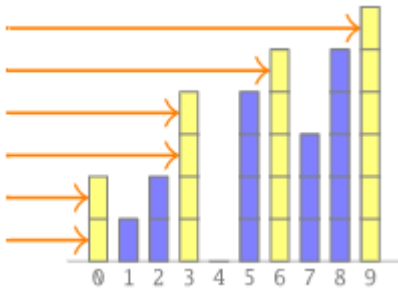
alors `indices(4, entiers)` renvoie `[]`

```
def indices(element, entiers):
```

```
    return liste
```

## 8 Soleil couchant sur les bâtiments

Lorsque des bâtiments sont alignés, ils se font de l'ombre les uns les autres. Dans cet exercice, nous sommes au soleil couchant, les rayons du soleil sont donc supposés horizontaux.



Le soleil couchant éclaire neuf bâtiments, les rayons du soleil sont représentés par des flèches horizontales.

- Les bâtiments aux indices 0 et 3 reçoivent des rayons de soleil alors que les bâtiments aux indices 1 et 2 sont masqués.
- Les **quatre** bâtiments aux indices [0, 3, 6, 9] reçoivent des rayons de soleil sur au moins un étage et sont donc éclairés, alors que les autres ne le sont pas.

Écrire une fonction `nb_batiments_eclaires` qui prend en argument la liste `hauteurs` des bâtiments et qui renvoie le nombre de bâtiments éclairés.

- La hauteur des bâtiments (en nombre d'étages) est donnée par une liste d'entiers positifs. Une hauteur de zéro étage signifie l'absence de bâtiment.

Pour l'exemple ci-dessus, cette liste est [2, 1, 2, 4, 0, 4, 5, 3, 5, 6].

### Exemple

Si `hauteurs = [2, 1, 2, 4, 0, 4, 5, 3, 5, 6]`  
alors `nb_batiments_eclaires(hauteurs)` renvoie 4

```
def nb_batiments_eclaires(hauteurs):
```

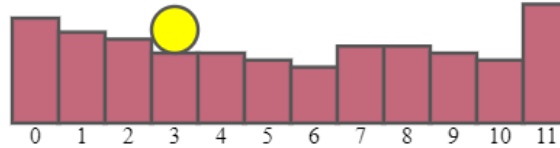
```
    return nombre_eclaires
```



## 9 Premier minimum local

Alors qu'elle joue sur un chemin dallé, Élodie laisse rouler une balle. En observant les dalles devant elle, elle se rend compte que certaines dalles sont plus basses que les précédentes, d'autres plus hautes.

Elle se pose la question suivante : "Où va s'arrêter la balle ?"



On donne les hauteurs des dalles dans le chemin sous la forme d'une liste de nombres entiers positifs. Cette liste contient au minimum deux valeurs. On garantit que la hauteur de la dernière dalle est strictement supérieure celles de toutes les autres. Dans l'exemple précédent, illustré par la figure, la balle s'arrête sur la dalle d'indice 6. En effet, la balle s'arrête sur **la première dalle dont la hauteur est strictement inférieure à celle de la suivante**.

On précise que lorsque deux dalles consécutives sont à la même hauteur, la balle continue de rouler.

Écrire une fonction `indice_arret` prenant en argument :

- Un tableau d'entiers nommé `hauteurs`

et renvoyant l'indice de la dalle sur laquelle s'arrête la balle. La balle est initialement sur la dalle d'indice 0.

### Exemples

Si `hauteurs = [3, 2, 5]`

alors `indice_arret(hauteurs)` renvoie `1`

Si `hauteurs = [3, 5]`

alors `indice_arret(hauteurs)` renvoie `0`

```
def indice_arret(hauteurs):
```

```
    return indice_arret
```

## 10 Double du précédent dans un tableau

Écrire une fonction `nombre_puis_double` prenant en argument :

- Un tableau d'entiers nommé `valeurs`

et renvoyant la liste de couples d'entiers  $(a, b)$  qu'il peut y avoir dans le tableau tels que  $b$  suit  $a$  et  $b$  est le double de  $a$ .

### *Exemples*

Si `valeurs = [7, 1, 2, 5, 3, 6]`

alors `nombre_puis_double(valeurs)` renvoie `[(1, 2), (3, 6)]`

Si `valeurs = [1, 4, 2, 5]`

alors `nombre_puis_double(valeurs)` renvoie `[]`

```
def nombre_puis_double(valeurs):
```

```
    return liste
```