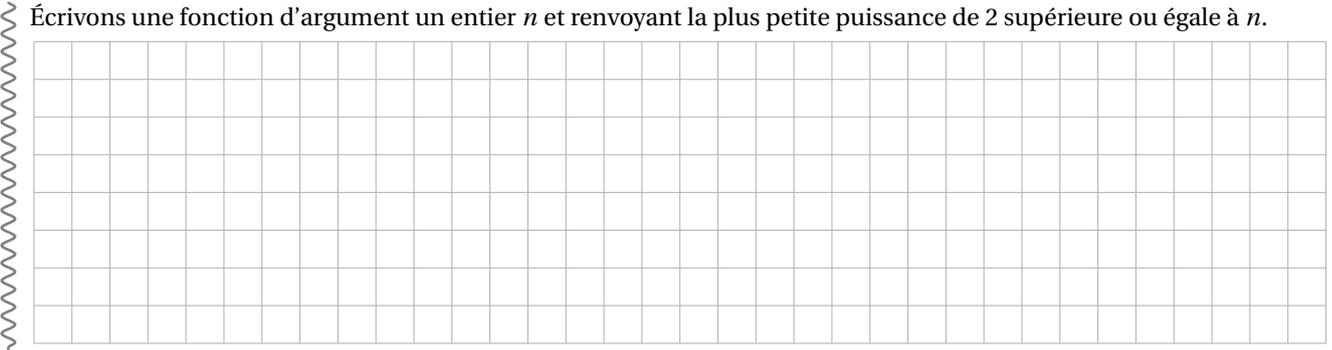




### Exemple

Écrivons une fonction d'argument un entier  $n$  et renvoyant la plus petite puissance de 2 supérieure ou égale à  $n$ .



### Exercice 1 (manipulation de boucles simples)

1. Écrire une fonction d'argument un entier  $n$  et renvoyant la somme des  $n$  premiers nombres impairs.
2. Écrire une fonction d'argument un entier  $n$  et renvoyant la somme des nombres impairs inférieurs ou égaux à  $n$ .
3. Écrire une fonction d'argument un entier  $n$  et renvoyant la valeur de factorielle de  $n$ .
4. Écrire une fonction d'argument un entier  $n$  supérieur ou égal à 2 et qui indique par un booléen si ce nombre est nombre premier.
5. Écrire une fonction d'argument une chaîne de caractères et renvoyant la longueur de cette chaîne de caractères (sans utiliser `len`).
6. Écrire une fonction d'argument une chaîne de caractères et renvoyant le nombre de voyelles (non accentuées) de cette chaîne de caractères.
7. Écrire une fonction d'argument une chaîne de caractères et renvoyant un booléen indiquant s'il s'agit ou non d'un palindrome.

### Exercice 2 (suite de Syracuse)

Pour tout entier naturel  $a$ , on peut considérer la suite  $(u_n)_{n \in \mathbb{N}}$  donnée par  $u_0 = a$  et pour tout  $n \in \mathbb{N}$  :

$$u_{n+1} = \begin{cases} \frac{1}{2}u_n & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

1. Écrire une fonction suite d'arguments deux entiers naturels  $a$  et  $n$  et qui renvoie le terme  $u_n$  de la suite correspondante.
2. Écrire une fonction attente d'argument un entier naturel  $a$  et qui renvoie le plus petit entier  $n$  tel que  $u_n = 1$  (pour la suite correspondante) ou qui renvoie  $-1$  si l'on n'a toujours pas rencontré la valeur 1 au bout de 1000 termes de la suite.

### Exercice 3

1. Écrire une fonction d'argument une liste d'entiers  $L$  et qui renvoie la liste formée par les valeurs positives de  $L$ .
2. Écrire une fonction d'argument une chaîne de caractères  $c$  et qui renvoie la liste formée par les indices pour lesquels il y a un e dans  $c$ .
3. Écrire une fonction d'argument une chaîne de caractères  $c$  et qui renvoie la liste formée du nombre de consonnes, du nombre de voyelles et du nombre d'autres caractères de  $c$  (on pourra supposer qu'il n'y a pas de lettre accentuée).

#### Exercice 4

Écrire une fonction suite d'arguments une fonction  $f$ , un nombre  $a$  et un entier  $n$ , et qui renvoie la liste des termes  $u_0, \dots, u_n$  de la suite  $u$  donnée par :

$$u_0 = a \text{ et } \forall n \in \mathbb{N}, u_{n+1} = f(u_n).$$

Pour tester cette fonction, on peut importer le module `math` à l'aide de l'instruction :

```
>>> import math
```

#### Exercice 5

Écrire une fonction `distance` prenant en argument une liste de nombres (de type `float`) qui calcule la somme des écarts entre les éléments de la liste et leur moyenne, c'est-à-dire que si la liste contient les nombres  $x_1, \dots, x_k$  alors la fonction doit renvoyer la valeur de :

$$\sum_{j=1}^k |x_j - \bar{x}| \text{ où } \bar{x} \text{ est la moyenne de } x_1, \dots, x_k.$$

#### Exercice 6

Écrire une fonction `separe` prenant en argument une liste de nombres entiers et qui renvoie un couple de listes : la première formée par les nombres pairs de la liste initiale et la seconde formée par les nombres impairs.

Par exemple, on doit obtenir :

```
>>> separe([2,3,5,8,7,9,9,4,11,-2,-1])
([2, 8, 4, -2], [3, 5, 7, 9, 9, 11, -1])
>>> separe([3,1,3,5,7])
([], [3, 1, 3, 5, 7])
```

#### Exercice 7 (recherche d'un maximum)

Considérons la fonction suivante :

```
1 def recherche(L):
2     """ entrée : L liste (non vide) de nombres
3         sortie : valeur maximale de L (nombre)
4     """
5     a = L[0]
6     for k in range(len(L)):
7         if L[k] > a:
8             a = L[k]
9     return a
```

1. Expliquer la ligne 4. Peut-on la remplacer par `a=0`?
2. Que se passe-t-il si, à la ligne 6, on remplace `L[k]>a` par `L[k]>=a`?
3. Proposer des modifications de la fonction ci-dessus pour obtenir une fonction renvoyant :
  - a. le couple formé par le minimum et le maximum des valeurs de  $L$ ;
  - b. le couple formé par le maximum de  $L$  et le nombre d'occurrences de ce maximum;
  - c. le couple formé par le maximum de  $L$  et la liste des indices de  $L$  en lesquels il y a cette valeur maximale;
  - d. la deuxième plus grande valeur de  $L$  (ou la plus grande si cette dernière est présente au moins deux fois).

### Exercice 8 (présence d'une valeur dans un tableau)

1. Écrire une fonction `presence` prenant en argument un tableau d'entiers et un entier et renvoyant un booléen indiquant si cet entier apparaît dans le tableau (sans utiliser `in`).

Par exemple, on doit obtenir :

```
>>> presence([1,2,3,4,5],3)
True
>>> presence([1,2,3,4,5],6)
False
```

2. Optimiser la fonction précédente dans le cas où le tableau d'entiers est trié dans l'ordre croissant.
3. Écrire une fonction `occurrences` prenant en argument un tableau d'entiers et un entier et renvoyant le nombre d'occurrences de cet entier dans le tableau.

Par exemple, on doit obtenir :

```
>>> occurrences([1,2,3,4,5],1)
1
>>> occurrences([1,2,3,1,1],1)
3
>>> occurrences([1,2,3,1,5],6)
0
```