

# Python (Listes)

Kais Klai

`kais.klai@lipn.univ-paris13.fr`

Dans ce qui suit, et pour chaque exercice, vous devez d'abord écrire l'algorithme avant de le traduire en Python.

1. Ecrire un programme qui remplit une liste de valeurs entières données par l'utilisateur. Le programme affiche ensuite la liste.
2. Ecrire un programme qui calcule et affiche le minimum d'une liste.
3. Ecrire un programme qui calcule et affiche la moyenne d'une liste.
4. Ecrire un programme qui cherche une valeur dans une liste. Si la valeur existe, la position de sa première occurrence sera affichée.
5. Ecrire un programme qui affiche le nombre d'occurrences d'une valeur donnée dans une liste.
6. Ecrire un programme qui insère une valeur dans une liste ordonnée (en la maintenant ordonnée).
7. Ecrire un programme qui vérifie si une liste est ordonnée (dans l'ordre croissant).
8. Ecrire un programme qui décale vers la gauche les valeurs contenues dans une liste.

Exemple :

– Avant 

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| D | E | C | A | L | A | G | E |
|---|---|---|---|---|---|---|---|

– Après 

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| E | C | A | L | A | G | E | D |
|---|---|---|---|---|---|---|---|

9. Etant donnée une liste d'entiers  $L$  telle que  $L[i] \in \{0, 1\}$ , écrire un programme qui affiche la position  $i$  telle que  $L[i]$  est le début de la plus longue série consécutive de zéros.
10. Ecrire un programme qui vérifie si deux listes contiennent les mêmes valeurs aux mêmes positions.
11. Ecrire un programme qui vérifie si deux listes contiennent les mêmes valeurs sans prendre en compte leurs positions, ni leurs nombres d'occurrences.
12. Ecrire un programme qui supprime la première occurrence d'une valeur, si elle existe, dans une liste.
13. Ecrire un programme qui supprime toutes les occurrences d'une valeur dans une liste.
14. Ecrire un programme qui calcule, dans une liste  $L_3$ , la fusion de deux listes ordonnées  $L_1$  et  $L_2$  telle que  $L_3$  est également ordonnée. Pensez à l'efficacité de votre algorithme.
15. Ecrire un programme qui calcule, dans une liste  $L_3$ , l'intersection de deux listes  $L_1$  et  $L_2$ . Nous supposons que chaque valeur apparaît une seule fois.
16. Ecrire un programme qui trie une liste de valeurs entières dans l'ordre croissant.

17. Ecrire un programme qui affiche le triangle de Pascal pour un nombre de lignes  $n$  donné.

**exemple pour  $n=5$**

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1

```

18. Ecrire un programme qui, pour une valeur  $n$  donnée, vérifie à la volée, si les  $n^2$  valeurs données par l'utilisateur forment un *carré magique*. Un carré magique est une matrice carrée dont la somme des lignes, des colonnes et des deux diagonales sont égales.

**exemple de carré magique pour  $n=3$**

|    |    |    |
|----|----|----|
| 14 | 5  | 17 |
| 15 | 12 | 9  |
| 7  | 19 | 10 |

19. Ecrire un programme qui, pour une valeur  $n$  donnée, vérifie à la volée, si les  $n^2$  valeurs données par l'utilisateur forment un *carré magique parfait*. Un carré magique est dit parfait si c'est un carré magique dont les valeurs sont comprises entre 1 et  $n^2$ .

**exemple de carré magique pour  $n=3$**

|   |   |   |
|---|---|---|
| 8 | 1 | 6 |
| 3 | 5 | 7 |
| 4 | 9 | 2 |

20. Ecrire un programme qui affiche le triangle suivant. Chaque ligne  $i$  est obtenue en comptant et en affichant le nombre d'occurrences de chaque valeur dans la ligne  $i - 1$ , suivie par la valeur elle-même.

```

1
1 1
2 1
1 2 1 1
1 1 1 2 2 1
3 1 2 2 1 1

```