

TP 2 : Rappels sur les listes

1 BCPST 2 - TP Informatique

CONSEIL : à chaque nouvelle question faites une nouvelle cellule en tapant "#%%" .
Vous pourrez ainsi traiter toutes les questions dans un même fichier.

Objectif du TP :- Savoir manipuler les listes.

- Connaître les opérations classiques sur les listes.
- Savoir coder une fonction de max, min, somme, moyenne des éléments d'une liste
- Revoir les listes de listes (les matrices)

I Exercice : Les listes simples

Q.1. Ecrire une fonction `somme(L)` qui prend en entrée une liste `L` et qui retourne la somme de ses éléments

Q.2. Ecrire une fonction `moyenne(L)` qui prend en entrée une liste `L` et qui retourne la moyenne de ses éléments.

Q.3. Ecrire une fonction `maximum(L)` qui prend en entrée une liste `L` et qui retourne son maximum.

Q.4. Ecrire une fonction `minimum(L)` qui prend en entrée une liste `L` et qui retourne son minimum.

Q.5. Ecrire une fonction `indice_maximum(L)` qui prend en entrée une liste `L` et qui retourne la position de son maximum.

```
L = [10,15,12,11]
print(indice_maximum(L) )
-----
-> 1 # car L[1] = 15 est le max
```

Q.6. Ecrire une fonction `indice_minimum(L)` qui prend en entrée une liste `L` et qui retourne la position de son minimum.

Q.7. Ecrire une fonction `second_maximum(L)` qui prend en entrée une liste `L` et qui retourne son 2eme plus grand élément.

```
L = [10,15,12,11]
print(second_maximum(L) )
-----
-> 12
```

Q.8. Ecrire une fonction `separe_liste(L)` qui prend en entrée une liste `L` et renvoie deux listes : la première et la deuxième moitié de `L`.

Exemple :

```
L = [1,2,1,3,1,5,2]
L1, L2 = separe_liste(L)
print(L1) ; print(L2)
-----
-> [1,2,1,3]
-> [1,5,2]
```

Q.9. Ecrire une fonction `indice(L,x)` qui prend en entrée une liste `L`, un élément `x` et qui retourne une liste contenant l'ensemble des positions auxquelles se trouve `x` dans la liste. Exemple :

```
L = [1,2,1,3,1]
L1 = indice(L,1)
print(L1)
-----
-> [0,2,4]
```

Q.10. Ecrire une fonction `supprime_doublons` qui prend en entrée une liste `L` et renvoie une liste qui contient les éléments de `L` uniques. (On pourra utiliser `nb_apparitions`)

Exemple :

```
L = [1,1,2,3,3,4,2,1]
L1 = supprime_doublons(L)
print(L1)
-----
-> [1,2,3,4]
```

Q.11. Ecrire une fonction `envers(L)` qui prend en entrée une liste `L` et qui retourne une liste `L1` qui contient les éléments de `L` dans l'ordre inverse :

```
L = [1,4,8,7,6]
print(envers(L))
-----
-> [6,7,8,4,1]
```

Q.12. Ecrire une fonction `alterne(L1,L2)` qui prend en entrée deux listes `L1` et `L2` de même taille et qui renvoie une liste dont les termes alterne entre ceux de `L1` et ceux de `L2`

```
L1 = [1,1,1] ; L2 = [2,3,4]
print(alterne(L1,L2))
-----
-> [1,2,1,3,1,4]
```

II Exercice : Les matrices dans python

Dans Python, on considère qu'une liste est une ligne et donc une matrice est une liste de lignes.

Pour représenter la matrice $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ On écrit :

```
M = [[1,2,3],[4,5,6],[7,8,9]]
```

Ainsi pour accéder au coefficient $m_{i,j}$ de la matrice, il suffit d'écrire :

```
print( M[2][3] )
-----
-> 6
```

Q.13. Créer une fonction `taille(M)` qui prend en entrée une matrice M et qui retourne son nombre de lignes et son nombre de colonnes.

Q.14. Créer une fonction `affiche(M)` qui prend en entrée une matrice M et qui affiche la matrice ligne par ligne

Exemple :

```
affiche( M )
-----
-> 1 2 3
   4 5 6
   7 8 9
```

Q.15. Créer une fonction `zero(n,p)` qui prend en entrée deux nombres entiers n et p et qui retourne la matrice nulle de taille $n \times p$.

Q.16. Créer une fonction `somme(A,B)` qui prend en entrée deux matrices A et B et retourne la matrice $A + B$.

Q.17. Créer une fonction `id(n)` qui prend en entrée une nombre entier n et qui retourne la matrice identité de taille $n \times n$

Q.18. Créer une fonction `transposee(M)` qui prend en entrée une matrice M et qui retourne la matrice tM

Q.19. Créer une fonction `produit(A,B)` qui prend en entrée deux matrices A et B et retourne la matrice AB .

Q.20. Créer une fonction `puissance(A,n)` qui prend en entrée une matrice A et un entier n et qui retourne la matrice A^n .

Q.21. Créer une fonction `bloc(A,B,C,D)` qui prend en entrée quatre matrices A, B, C, D et un entier n et qui retourne la matrice $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$

Q.22. Coder un algorithme créant les matrices suivantes :

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

adapter le code pour construire de telles matrices de taille $n \times n$.

Q.23. Construire la matrice suivante de taille $n \times n$ (ici $n = 7$) :

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 14 & 13 & 12 & 11 & 10 & 9 & 8 \\ 15 & 16 & 17 & 18 & 19 & 20 & 21 \\ 28 & 27 & 26 & 25 & 24 & 23 & 22 \\ 29 & 30 & 31 & 32 & 33 & 34 & 35 \\ 42 & 41 & 40 & 39 & 38 & 37 & 36 \\ 43 & 44 & 45 & 46 & 47 & 48 & 49 \end{pmatrix}$$