

Vos réponses aux questions devront être enregistrées dans le fichier TP2.py disponible sur l'ENT. Ce fichier est à déposer sur la zone de dépôt à la fin du TP.

### Introduction à la notion de fonction

Nous allons commencer à programmer des fonctions avec Python. En informatique, une fonction prend en paramètres des arguments et retourne un résultat. La déclaration en python se fait avec le mot `def` sous la forme suivante :

```
def f(arguments):  
    [instructions]  
    return resultat
```

Par exemple la fonction `somme` qui suit, réalise la somme de deux entiers :

```
def somme(x,y) :  
    return x+y
```

On récupère le résultat d'une fonction avec une instruction d'affectation, par exemple `s=somme(3,4)` affecte à `s` la valeur 7.

**Exercice 1.** Sans utiliser les fonctions `abs`, `pow` ou encore l'opérateur `**` :

1. Définir une fonction `absolue` prenant en entrée un entier et retournant sa valeur absolue. (On ne demande pas de vérifier que le paramètre passé à la fonction est un entier).
2. Définir une fonction `fact` prenant en entrée un entier positif et retournant sa factorielle définie par  $n! = \prod_{i=1}^n i = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$ .
3. Définir une fonction `puissance` prenant deux arguments  $x$  et  $n$  et retournant  $x^n$ . (On supposera que  $n$  est un entier positif, et on posera  $x^0 = 1$  pour tout réel  $x$ ).

### Application aux matrices

**Définition 1.** Si  $M = (a_{i,j})$ , on note  ${}^tM = (m_{j,i})$  la **matrice transposée** de  $M$ .

**Exemple 1.** Si  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$  alors  ${}^tA = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$ .

**Exercice 2.** Dans chacun cas suivant, écrire une fonction qui résout le problème posé sans utiliser les fonctions prédéfinies de python.

1. Écrire une fonction `transpose` qui prend en entrée une matrice  $A$  de taille  $p \times n$  et retourne une matrice  $B$  égale à la transposée  ${}^tA$  de  $A$ .
2. Écrire une fonction `diagonale` qui prend en entrée une matrice  $A$  de taille  $p \times n$  et retourne une matrice  $B$  de même taille dont les coefficients de la diagonale sont égaux à ceux de  $A$  et les autres sont nuls.
3. Écrire une fonction `trace` qui prend en entrée une matrice  $A$  de taille  $n \times n$  et retourne la somme des valeurs de la diagonale principale.
4. Écrire une fonction `valeurmax` qui prend en entrée une matrice  $A$  de taille  $n \times p$  quelconque et retourne la valeur maximale des coefficients de  $A$  et sa position  $(i\_max, j\_max)$  dans la matrice  $A$ .