


## TP3 : les boucles

**Objectif :** maîtriser l'utilisation des boucles avec Python.

 **N.B** les exercices de ce TP sont déjà abordés durant les séances TD, donc cela suppose que vous avez déjà la solution algorithmique !


### Exercice 1 : Choix de la structure itérative

Pour chacun des problèmes suivants, proposer un programme Python en choisissant la structure itérative la mieux appropriée (for ou while) :

- Calcul de la somme des N premiers entiers naturels avec N saisi par l'utilisateur.
- Boucle d'attente qui se termine si l'utilisateur entre la valeur 100
- Lecture de 10 valeurs réelles comprises entre 0 et 20 (si ce n'est pas le cas, il faut redemander la valeur à l'utilisateur) ; puis afficher :
  - Le nombre de valeurs inférieur strictement à 10
  - Le nombre de valeurs supérieur à 10 et inférieur strictement à 15
  - Le nombre de valeurs supérieur à 15
- Calcul et affichage du plus grand nombre N tel que  $\sum_{i=0}^N i \leq S$  avec S un nombre supérieur à 1 saisi par l'utilisateur.  
Exemple : si l'utilisateur donne la valeur 10 pour S, alors le nombre N recherché est 4 (0+1+2+3+4 ≤ 10 par contre 0+1+2+3+4 +5 =15 > 10)

### Exercice 2 : Compte à rebours

Ecrire un programme Python qui demande à l'utilisateur de saisir un nombre entier n positif, puis afficher tous les nombres par ordre décroissant entre n et 0. Proposer deux versions, une avec la boucle 'for' et l'autre avec la boucle 'while'.

 Vous pouvez importer le module `time` et utiliser la méthode `sleep(0.2)` de ce module pour avoir un effet compte à rebours !

**Point d'étape 1 à faire valider**

### Exercice 3 : Jeu du nombre mystère

Ecrire un programme Python qui tire une valeur aléatoire x entre 0 et 100 et qui demande à l'utilisateur de deviner cette valeur. L'algorithme répond à chaque fois si la valeur est plus

grande, plus petite ou égale à la valeur à deviner. Un compteur incrémenté à chaque passage donnera le nombre de tours qui ont été nécessaires pour trouver le nombre mystère.

Vous pouvez utiliser le module **random** vu en TP1 !

## Exercice 4 : Factorielle itérative

Ecrire un programme Python qui détermine la factorielle d'un entier n (donné par l'utilisateur) avec les deux types de boucles. Votre programme doit permettre à l'utilisateur de choisir la boucle à utiliser puis réalise le traitement nécessaire en conséquence.

Votre programme doit permettre aussi l'affichage de l'évolution de la valeur de la factorielle à chaque itération !

*Point d'étape 2 à faire valider*