



---

*L'objectif de ce TP est de maîtriser l'utilisation des structures itératives en Python.*

---

**Exercice 1.** Écrivez un programme qui demande un entier  $n$  et affiche tous les entiers de 0 à  $n$ .

**Exercice 2.** Écrire un programme qui saisit une chaîne de caractères au clavier et l'affiche à l'envers.

**Exercice 3.** Écrire un programme qui affiche la table des codes de l'ASCII des caractères compris entre 32 et 127. De même pour les codes de l'Unicode des caractères compris entre 1567 et 1624.

**Exercice 4.** Écrire un algorithme et sa traduction en un programme Python qui demande un entier  $n > 0$  et dessine  $n$  lignes. Par exemple pour  $n = 5$  :

```
*  
***  
*****  
*****  
*****
```

**Exercice 5.** Écrire un programme qui demande la saisie d'une chaîne d'ADN, compte le nombre total de ses bases, puis le nombre de chacune des bases.

**Exercice 6.** Écrire un script qui convertit les entiers positifs en base binaire .

**Exercice 7.** Écrire un programme Python qui calcule le pgcd de deux entiers en utilisant l'algorithme d'Euclide :

```
// calcul du pgcd de a et b  
repete  
    reste <-- le reste de la division de a par b  
    a <-- b  
    b <-- reste  
jusqu'a ce que reste=0  
pgcd <-- a
```

**Exercice 8.** Écrire un algorithme et sa traduction en un programme Python qui demande la saisie d'une chaîne d'ADN, compte le nombre total de ses bases, puis le nombre de chacune des bases.

**Exercice 9.** Écrire un algorithme et sa traduction en un programme Python qui demande à l'utilisateur de taper un mot de passe à stocker dans une variable. Si le mot de passe est erroné 3 fois de suite le programme affiche : "Carte bancaire retenue" sinon il affiche "mot de passe correcte".

**Exercice 10.** Écrire un algorithme et sa traduction en un programme Python qui invite l'utilisateur à entrer une séquence d'ADN et la transforme en son complémentaire (A devient T, T devient A, C devient G et G devient C). Le programme affichera la séquence entrée et en dessous la séquence transformée.

**Exercice 11.** Écrire un algorithme et sa traduction en un programme Python qui invite l'utilisateur à entrer une séquence d'ADN et calcule le pourcentage de GC dans cette séquence.