



---

*L'objectif de ce TP est de maîtriser les fonctions et les fichiers en Python 3.*

---

**Exercice 1** Écrire une fonction qui affiche la parité d'un nombre entier.

**Exercice 2** Écrire une fonction appelée `est_adn` qui prend comme argument une chaîne de caractère et retourne 1 si la chaîne est une chaîne d'ADN, et 0 sinon.

**Exercice 3** Écrire une fonction appelée `PM_ADN` qui calcule le poids moléculaire d'une chaîne d'ADN. Sachant que le poids d'un couple AT est d'environ  $260 \mu\text{g}/\mu\text{mol}$ , et le poids d'un couple GC d'environ  $245 \mu\text{g}/\mu\text{mol}$ .

**Exercice 4** Écrire une fonction appelée `Cherche_TATA` qui recherche une boîte TATA dans une chaîne d'ADN et retourne l'indice de la base où commence la boîte TATA, ou -1 s'il n'y a pas de boîte TATA.

**Exercice 5** Écrire un script en Python qui compte le nombre de A, de C, de G et de T d'une chaîne d'ADN.

**Exercice 6** Écrire une fonction récursive qui calcule le PGCD de deux nombres par la méthode d'Euclide.

**Exercice 7** Écrire en Python un programme qui :

- demande à l'utilisateur un nom de fichier,
- ouvre un fichier qui porte ce nom,
- demande à l'utilisateur ce qu'il veut écrire dans ce fichier ?
- écrit ce texte saisi par l'utilisateur dans ce fichier,
- affiche à l'utilisateur "Votre texte est sauvegardé."

**Exercice 8** Écrire en Python un programme qui met la table de multiplication dans un fichier.

**Exercice 9** Recopier un fichier texte en omettant toutes les lignes qui commencent par un caractère `#`.

**Exercice 10** Écrire une fonction qui résout les équations  $aX^2 + bX + c = 0$  et sauvegarde les solutions de toutes les équations résolues dans un fichier, sous la forme :

les solutions de l'équation  $X^2+2X+1=0$  sont ...

les solutions de l'équation  $X^2+X+1=0$  n'existent pas dans  $\mathbb{R}$  etc ...

**Exercice 11** Écrire une fonction en Python qui copie et numérote dans un autre fichier toutes les lignes d'un fichier.

**Exercice 12** Écrire une fonction en Python qui copie dans un autre fichier toutes les lignes paires d'un fichier.

# 1 test: Solutions

## 1.1 Some Solutions

*Solution 1* def Parite(nombre):

```
    """ Affiche la parité d'un nombre entier """
    if (nombre % 2) == 1:
        print(nombre,'est impaire')
    else :
        print(nombre,'est paire')
```

*Solution 2* def est\_adn(chaine):

```
for base in chaine:
if (base != "a") and (base != "t") and (base != "c") and (base != "g"):
return 0
return 1
```

*Solution 3* def PM\_ADN(adn):

```
    poids = 0.0
    for base in adn:
if (base == "A") or (base == "T"): poids = poids + 260.0
elif(base == "C") or (base == "G"): poids = poids + 245.0
return poids
```

*Solution 4* def cherche\_TATA(adn):

```
for i in range(len(adn) - 3):
if ((adn[i] == "T") and (adn[i + 1] == "A") and
    (adn[i + 2] == "T") and (adn[i + 3] == "A")):
return i
return -1
```

*Solution 5*

*Solution 6* def pgcd ( p , q ) :

```
p , q = max ( p , q ) , min ( p , q )
if q == 0 :
return p
else :
return pgcd ( q , p % q )
```

*Solution 7*

*Solution 8* # table de multip dans un fichier

```
obj=open('rr', 'w')
```

```
for i in range(1,11):
```

```
    obj.writelines(['table de multiplication de',str(i),'\n'])
```

```
    for j in range(1,11):
```

```
obj.writelines([str(i), 'x', str(j), '=', str(i*j), '\n'])
obj.close()
```

*Solution 9* # Supprime les lignes commençant par #

```
def filtre(source, destination):
    """ Recopier un fichier en éliminant les lignes de remarques """
    fs = open(source, 'r')
    fd = open(destination, 'w')
    while 1:
        txt = fs.readline()
        if txt == '':
            break
        if txt[0] != '#':
            fd.write(txt)
    fs.close()
    fd.close()
    return
```

*Solution 10*

*Solution 11*

*Solution 12*

---

## 1.2 Extra Solutions