

# Initiation à la programmation – exos notés à rendre à la fin du TP8 – 11 octobre 2023

Vous devez rendre votre travail sur ecampus avant la fin du TP soit 15h30. Vous devez être présent dans la salle pour être noté (tous les autres auront 0).

Vous allez donc commencer par créer un dossier qui s'appelle nom\_prenom . On mettra ensuite dans ce dossier les fichiers .py demandés. Puis vous rendrez ce dossier après l'avoir zippé.

## Exercice 1

1. Ecrire la fonction `divise(p,n)` où `n` et `p` sont des entiers, qui teste si `p` divise `n`. On doit renvoyer `True` ou `False`. On utilisera `n%p == 0`.
2. Ecrire la fonction `lesdiviseurs(n)` qui teste les entiers de 1 à `n-1` (inclus) et renvoie la somme des entiers qui divisent `n`.
3. Ecrire la fonction `parfait(n)` qui renvoie `True` si `n` est un entier parfait et `False` sinon. Un entier est parfait s'il est égal à la somme de ses diviseurs (sauf lui).  
6 est parfait car  $6 = 1 + 2 + 3$   
24 n'est pas parfait car  $24 \neq 1 + 2 + 3 + 4 + 6 + 8 + 12$
4. Ecrire la fonction `afficheParfait()` sans paramètre, qui demande un entier `nb` à l'utilisateur et affiche successivement tous les entiers parfaits entre 2 et `nb` (compris) (voir exemple ci-dessous).

```
>>> somme((2,8,1,0,2))
13
>>> divise(3,17)
False

>>> divise(3,15)
True

>>> lesdiviseurs(24)
36
>>> lesdiviseurs(6)
6
>>> lesdiviseurs(10)
8


>>> parfait(6)
True

>>> parfait(24)
False

>>> afficheParfait ()
Entrez un nombre: 1000
Les entiers parfaits entre 2 et 1000 sont :
```

## Exercice 2

On trouvera dans le fichier `donnees.py` des tuples pour cet exercice. On considère des tuples de mots, chaque mot étant une chaîne de caractères.

```
groupe1=("lena", "anna", "luc", "raphael", "ali", "ben", "leo", "isabella", "manon",  
"pierre", "mohamed", "luke", "loris", "louana")  
groupe2=("anna", "pierre", "jean", "david", "guillaume", "mohammed", "lucia", "leo")
```

1. Ecrire la fonction `CompteVoyelles(m)` qui renvoie le nombre de voyelles présentes dans le mot `m` placé en paramètre.
2. Ecrire la fonction `affiche(t)` qui étant donné un tuple de mots `t`, affiche successivement tous les prénoms contenant exactement 3 voyelles.
3. Ecrire la fonction `commun(t1,t2)` qui étant donné deux tuples `t1` et `t2` renvoie le tuple des éléments communs aux deux tuples.

```
>>> CompteVoyelles(" r a p h a e l ")  
3  
  
>>> CompteVoyelles(" i s a b e l l a ")  
4  
  
>>> affiche(groupe1)  
raphael  
pierre  
mohamed  
  
>>> commun(groupe1,groupe2)  
( 'anna', 'leo', 'pierre' )
```