

**TP3 - Chaînes de caractères et tuples**

- Q.1) Récupérez le fichier `TP3.py` sur cahier de prépas.
- Q.2) Ouvrez pyzo puis le fichier `TP3.py`.
- Q.3) Corriger le code proposé pour qu'il corresponde à la spécification donnée.
- Q.4) Écrire une fonction `carre(n)` qui affiche un carré de taille `n` constitué de `*`. Par exemple, `carre(4)` doit afficher :

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

- Q.5) Écrire une fonction `pyramide(n)` qui affiche pour `n = 4` :

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

- Q.6) Écrire une fonction `identiques(t)` qui renvoie `True` si tous les éléments du tuple `t` sont identiques et `False` sinon.
- Q.7) Écrire une fonction `nb_mots(texte)` qui renvoie le nombre de mots dans `texte`.
- Q.8) Écrire une fonction `nb_lettres(chaine)` qui renvoie le nombre de lettres dans `chaine`. On suppose que `chaine` n'est constituée que de lettres, d'espaces, de virgules, de points et d'apostrophes.
- Q.9) Écrire une fonction `occurences(caractere, chaine)` qui renvoie le nombre de fois que la lettre représentée par `caractere` apparait dans `chaine`.  
Pour gérer les majuscules, on pourra utiliser `chaine.lower()` qui renvoie une copie de la chaîne initiale où toutes les majuscules sont remplacées par des minuscules.
- Q.10) Écrire une fonction `somme(n)` qui prend en argument un entier et renvoie la somme de ses chiffres. Par exemple :

```
>>> somme(2022)
6
```

On pourra commencer par convertir `n` en chaîne de caractères.

- Q.11) Écrire une fonction `nb_lettres2(texte)` qui affiche successivement le nombre de lettres dans chaque mot de `texte`.