Introduction à la programmation (C5-101115-INFO)
Licence 1 - Année 2022/2023
TP 8 - Listes et tuples (suite)

B. Besserer, C.Demko, P. Franco, L. Mascarilla, C. Saint-Jean, E. Zahzah

En préambule du TP 8, avez-vous des questions sur le TP7 ?

Exercice 1: *Liste d'emails (*mels.py*)*

1. Après avoir récupéré le fichier *emails.txt* sur Moodle, copier-collez le code suivant :

```
with open('emails.txt') as f:
    M = f.read().splitlines()
```

   puis afficher M.
2. Combien d'adresses contient *M* ?
3. Quelle est l'adresse email la plus longue ?
4. À l'aide de la méthode *split* (cf. help(str.split)), construisez la liste *D* des noms de domaine des adresses emails.

   `president.salengro@présipauté.gro ---> présipauté.gro`
5. Créer la liste *Du* analogue à *D* en supprimant les doublons.
   Indication : 'in' et 'not in' permettent de tester l'appartenance ou non à une liste.
6. Compter le nombre de personnes par nom de domaines. On fait ici l'hypothèse les personnes sont regroupées par nom de domaine.
7. Construire la liste contenant l'ensemble de tuples :

   (<nom de domaine>, <nombre d'adresses pour ce nom de domaine>)

(Bonus) Compter le nombre d'adresses dont le domaine termine par ".fr"

Exercice 2: *Liste des nombres premiers (*nombres_premiers.py*)*

On rappelle qu'un nombre n est premier si et seulement si n a pour seuls diviseurs 1 et lui-même. Par convention, 1 n'est pas premier. Par exemple, 7 est premier car 2, 3, 4, 5, 6 ne divisent pas 7.

1. À l'aide d'une boucle *for*, montrer que 97 est premier.
2. En réutilisant ce code dans le parcours des entiers entre 2 et 1000, construire la liste P des nombres premiers inférieurs à 1000.
3. Combien y en a t'il ?

Si l'on parcourt les entiers de 2 à 1000 (variable n), par ordre croissant, la liste P en construction contient l'ensemble des nombres premiers inférieur à n.
Il semble alors judicieux de limiter le test de divisibilité aux seuls de éléments de P.
Par exemple, pour n=7, on testerait 2, 3, 5 au lieu de 2, 3, 4, 5, 6.

1. Écrire une nouvelle version de votre programme pour appliquer ce principe.
2. A t'on un gain de rapidité ?

Une autre stratégie est de parcourir la *liste* des entiers de 2 à 1000 en éliminant les multiples d'un nombre premier. En détails :

- Au départ L = [2, 3, 4, 5, 6, 7, 8, ..., 1000]
- L[0] = 2 est premier, on l'ajoute à la liste des nombres premiers P.
- On élimine 2 (=1*2), 4 (=2*2), 6 (=3*2), 8, 10 ,12 ..., 1000 de L
- L[0] = 3 est premier, on l'ajoute à P.
- On élimine 3 (=1*3), 6 (=2*3), 9 (=3*3), 12, 15, ... 999 de L
- L[0] = 5 est premier, on l'ajoute à P.
- On élimine 5 (=1*5), 10 (=2*5), ... de L
- On continue jusqu'à que L soit vide.

1. Écrire une nouvelle version de votre programme pour appliquer ce principe.
2. Est ce plus lent ? si oui, pourquoi ?