Matière : Programmation Python **Auditoire : PLSI** **Responsable de cours :** Ikram Chaâbane	**Année universitaire 2022-2023**

TP n° 2 : les séquences chaines – listes - tuples

Objectifs

1. *Manipulation des séquences en Python*
2. *Savoir choisir la structure adéquate à un problème donné*

Exercice 1

Ecrire un script python qui permet de saisir une phrase du clavier et:

1- afficher son nombre de mots
2- supprimer les espaces **superflus**
3- afficher le nombre **d'occurrence** du caractère « a » dans cette phrase
4- afficher la phrase dans un ordre renversé de **caractères**
5- afficher la phrase dans un ordre renversé de **mots**
6- vérifier si **tous** les mots commencent par une voyelle
7- remplacer **toutes** les occurrences de « er » par « ons »
8- remplacer la **1ère occurrence** de « amis » par « comptes »
9- afficher la phrase saisie toute en commençant par une **majuscule chaque mot**
10- afficher les mots de la phrase dans l'ordre **décroissant** de **longueur de mots**
11- trier cette phrase dans l'ordre alphabétique croissant de la **2e lettre de chaque mot**

Si la phrase

Exemple :

```
Saisir une phrase:      les ber amis font de ber      amis
1: 7
2: les ber amis font de ber amis
3: a se répète 2 fois
4: sima reb ed tnof sima reb sel
5.1: amis ber de font amis ber les
5.2: amis ber de font amis ber les
6: non, mot les invalide
7: les bons amis font de bons amis
8: les bons comptes font de bons amis
9: Les Bons Comptes Font De Bons Amis
10: ['comptes', 'bons', 'font', 'bons', 'amis', 'les', 'de']
11: ['les', 'de', 'amis', 'bons', 'comptes', 'font', 'bons']
```

Exercice 2

Ecrire un programme Python qui lit deux mots (**mot** et **mot_c**) et vérifie si le mot **mot** est composable à partir du mot **mot_c**

Exemples

- **mot** = "python", **mot_c** = "aophrtkny", le programme affiche [python] est composable à partir de [aophrtkny]

- **mot** = " python", **mot_c** = "miduyhnq ", le programme affiche [python] n'est pas composable à partir de [miduyhnq]

NB. Un mot est composable à partir d'une séquence de lettres si la séquence contient toutes leslettres du mot. Chaque lettre de la séquence ne peut être utilisée qu'une seule fois

Exercice 3

Afficher et interpréter les résultats des instructions suivantes :

1- print([x ** 2 for x in [1, 2, 3, 4]])
2- print([i for i in range(10) if i%2==0])
3- print([i for i in range(50) if i%2==0 if i%3==0 if i%9==0])
4- print(["Python" if i%3==0 else "C" for i in range(2,10)])
5- print([(i,j) for i in range (5) for j in range (2)])
6- list = [[2,4,6,8]]

```
matrix = [[row[i] for row in list] for i in range(4)]
print(matrix)
```

Exercice 4

Ecrire un programme python qui permet de saisir deux entiers **m** et **n**, et une liste **lis** de **m** × **n** nombres entiers. Le programme doit construire à partir de la liste une matrice **m** × **n** et l'afficher.

Exemple : si **m** =3, **n** =2 et **lis** = [0,1,0,0,1,1] alors le programme doit afficher [[0,1],[0,0],[1,1]]

Exercice 5

Ecrire un programme python qui permet de :

1- lire une liste de mots. La lecture s'arrête lorsque l'utilisateur tape la lettre Q.
2- construire une liste de tuples où chaque tuple contient un mot de la liste et sa longueur. Utiliser les listes de compréhension
3- afficher l'indice du tuple associé au mot le plus court
4- afficher la racine des mots lus s'il existe, sinon le programme affichera « pas de racine commune ». La racine des mots représente le préfix commun de tous les mots de la liste.

Exemple d'exécution :

1-

```
lire mot num 1: programming
lire mot num 2: program
lire mot num 3: programmer
lire mot num 4: programs
lire mot num 5: programmers
lire mot num 6: q
['programming', 'program', 'programmer', 'programs', 'programmers']
```

2-

```
[('programming', 11), ('program', 7), ('programmer', 10), ('programs', 8), ('pr
ogrammers', 11)]
```

3-

1 est l'indice du mot le plus court

4- racine= program

Bon travail