Travaux pratiques n°2	#Conteneurs standards en Python
Matière : Programmation Python	**Enseignants :** M. Taoufik Ben Abdallah
Discipline : 1ère année Génie Informatique	Mme. Fatma Ben Said
Année Universitaire : 2024-2025 / S1	

Exercice 1

Étant donnée la chaine de caractères **mot=" Aap44193th23zyt889D3on "**

1/ Supprimer les espaces de bord de **mot**

>mot=Aap44193th23zyt889D3on

2/ Afficher le nombre de lettres en majuscule dans **mot**

>Nombre de lettres en majuscule=2

3/ Afficher le nombre de caractères **non redondants** (les caractères uniques) dans **mot**

>Nombre de caractères non redondants=11

4/ Dans une seule ligne de code, remplacer les caractères **"A"** ou **"a"** de **mot** par **"?"**, inverser la chaine, et afficher la nouvelle chaine de caractères obtenue

>mot=no3D988tyz32ht39144p??

5/ Remplacer les sous-chaines composées de trois chiffres successifs dans **mot** par le symbole**"@"**

>mot=no3D@tyz32ht@44p??

6/ Calculer la somme des chiffres restants qui apparaissent dans **mot**

>Somme des chiffres=16

7/ Vérifier si **mot** est composable à partir de la chaine de caractère **"Ppython"**. **La vérification ne doit pas être sensible à la casse**

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

>no3D@tyz32ht@44p?? est non composable de PPYTHON

Exercice 2

Étant donnée la chaine de caractères **ch** suivante :

```
ch="Python est est est populaire \n Python populaire est simple"""
```

1/ Créer **une liste**, nommée **l_ch**, contenant des listes dont chacun comporte **deux éléments** : le premier représente un mot de **ch** et le deuxième correspond à son **nombre d'occurrence**

>l_ch= [['Python', 2], ['est', 4], ['populaire', 2], ['simple', 1]]

2/ Mélanger **aléatoirement** les éléments de **l_ch**

>Exemple de l_ch= [['Python',2], ['populaire',2], ['est',4], ['simple',1]]

3/ Transformer les listes de format **[mot , occ]** de **l_ch** en des listes contenant **mot** répété **occ** fois

>l_ch= [['Python', 'Python'], ['populaire', 'populaire'], ['est', 'est', 'est', 'est'], ['simple']]

4/ Transformer **l_ch** en une chaine de caractères, nommée **chT** dont les mots sont séparés par **"*"**

>chT= "Python*Python*populaire*populaire*est*est*est*est*simple"

Exercice 3

1/ Saisir un entier positif **n** ≤**20**, puis **générer aléatoirement** un tuple, nommé **tupp**, de **n** entiers qui varient entre **1** et **10**. **Répéter la saisie** de **n** **si la valeur donnée est erronée**

>Exemple de tupp= (2,2,1,3,4,6,1)

2/ Extraire à partir de **tupp** un tuple, nommé **tupp_seg**, qui contient **m** listes dont chacun comporte **k** chiffres ordonnés dans l'ordre décroissant

>tupp_seg= ([2,2,1],[3],[4],[6,1])

3/ Ajouter à la première position de chaque liste de **tupp_seg** une valeur représentant la somme de ses éléments

>tupp_seg= ([5,2,2,1],[3,3],[4,4],[7,6,1])

4/ Écrire en **une seule instruction** un programme permettant de transformer **tupp_seg** en une chaîne de caractères, nommée **ch_nombre**, où **chaque nombre dupliqué n'apparaît qu'une seule fois**

>ch_nombre= "5213461"

Exercice 4

Étant donné le dictionnaire **b_pharm** qui contient **les produits pharmaceutiques**, où **les noms sont utilisés comme clés** et **les prix unitaires sont donnés en tant que valeurs correspondantes**.

```
b_pharm={'Doliprane':3.000, 'Efferalgan': 4.500, 'orelox': 9.250, 'EcranFacial':35.900,
'Paracétamol': 11.800, 'Atropine': 22.500, 'Augmentin': 23.400,'GelNettoyant':15.500,
'Thermomètre':8.200, 'SondeAspiration':12.500}
```

Un client peut commander un ou plusieurs médicaments. Sa commande est stockée dans une **liste de dictionnaires**, nommée **c_client**. Chaque produit dans la commande est représenté par un dictionnaire où **la clé représente le nom du produit** et **la valeur représente la quantité**. Dans cet exercice, en supposant que **c_client** est défini comme suit : **[{'Doliprane': 1}, {'GelNettoyant': 2}, {'Dermosalic': 1}, {'Fervex': 3}]**

1/ Transformer **en une seule instruction** **c_client** en un dictionnaire de la forme suivante : **{'Doliprane': 1, 'GelNettoyant': 2, 'Dermosalic': 1, 'Fervex': 3}**

2/ Créer et afficher un **ensemble** nommé **prod_disp** contenant les produits du client **c_client** qui **sont disponibles** dans la pharmacie **b_pharm**

3/ Créer et afficher l'**ensemble** des **produits non disponibles** dans la commande du client, nommé **prod_n_disp**, **sans recourir à une structure itérative ou à une écriture en compréhension**

4/ Traiter la commande du client en transformant **c_client** en une **liste**, **l_cmd**, **contenant un tuple de 2 éléments pour chaque produit dans la commande**. Le premier élément représente le nom du produit commandé, et le deuxième représente le **prix total** s'il est disponible, sinon il est défini comme **None**. Le résultat sera le suivant : **[('Doliprane', 3.0), ('GelNettoyant', 31.0), ('Dermosalic', None), ('Fervex', None)]**

5/ Déterminer et afficher **en une seule instruction** le montant total à payer de la commande **c_client**. Si le nombre de produits non disponibles dans la commande est supérieur à **1**, appliquer une réduction de **10%**

Bon Travail ♣