Exercices de révision Python

Exercice 1

1. Écrire un programme python qui permet de saisir une suite de nombre (compris entre 100 et -100) tant que l`utilisateur saisit un nombre diffèrent de 0. Le programme doit afficher

* Le nombre d`entier négatif
* Le nombre d'entier positif
* La somme et la moyenne des nombres saisis
* Une liste LP qui contient tous les nombres positifs
* Une liste LN qui contient tous les nombres négatifs

1. Soient les listes suivantes : t1 = [31,28, 31,30, 31, 30, 3I,31, 30, 31, 30, 31]t2 = ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin', 'Juillet’, 'Août’, 'Septembre', 'Octobre', 'Novembre', 'Décembre']Écrire un petit programme qui crée une nouvelle liste t3, celle-ci devra contenir tous les éléments des deux listes en les alternant, de telle manière que chaque nom de mois soit suivi du nombre de jours correspondant : ['Janvier',31,'Février',28,'Mars',31, etc…]\_
2. Qu`affiche le programme ci-dessous :L= [ i for i in range (20)]print (L)print ( [ i for i in range (20) if i%2==1])print([i\*i for i in range(1,6)])print ([ i+ j for i in range (1,6) for j in range (1,6)])

Exercice 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lors d’un TP dont le but est de déterminer la relation entre la masse et le poids d'un objet, un groupe d’étudiants a obtenu les mesures suivantes. | masse (en kg) | 0.01 | 0.05 | 0.1 | 0.2 | 0.5 |
| Poids (en N) | 0.1 | 0.5 | 0.95 | 2.1 | 5 |

1. Créer une liste « masse ›› contenant les valeurs du tableau de résultat.
2. Créer une liste « poids ›› contenant les valeurs du tableau de résultat.
3. Écrire un programme permettant de calculer le rapport P/m pour chaque mesure et insérer les valeurs obtenues dans une liste nommée « rapport »
4. Définir une fonction multiplier (Liste\_Poids, K) qui multiplie chaque élément de la liste «Liste-Poids » par le nombre K.
5. Déﬁnir une fonction Sup (Liste-Masse, M) qui renvoi la liste de masses supérieures strictement à M.
6. Définir une fonction permettant de trier une liste par ordre croissant selon un algorithme de tri par sélection

Rappel : Sur une liste de n éléments, le principe du tri par sélection est le suivant :

* Rechercher le plus petit élément de la liste et le changer avec l’élément d’indice 0 ;
* Rechercher le second plus petit élément de la liste et l’échanger avec l’élément d’indice 1
* Continuer de cette façon jusqu`à ce que la liste soit entièrement triée.

Exercice 3 :

1. Afficher chaque caractère d’une chaîne en utilisant une boucle For.
2. Écrire une fonction nbMots qui prend une chaîne de caractères et renvoie le nombre de mots dans la phrase. Par exemple. nbMots(`Bonjour comment ça va ?’) renverra 4. Les mots sont simplement les blocs de caractères délimité par les espaces.
3. Écrire une fonction nbre0currences(mot,x) qui prend en argument une chaîne de caractère mot et un caractère x et qui renvoie le nombre de fois où le caractère x est présent dans mot.
4. Définir une fonction inverse(chaine) qui permet d`inverser l’ordre des caractères d’une chaîne quelconque. La chaîne inversée sera renvoyée au programme appelant.

Exercice 4 : On considère le dictionnaire suivant dont les clés sont les noms des élèves et les valeurs des clés sont les moyennes générales obtenues en passant l’examen final.

Etudiants = {"etudiant\_1" : 13 , "etudiant\_2" :17 ,"etudiant\_3" : 9, "etudiant\_4" : 15 , "etudiant\_5" : 8,

"etudiant\_6" : 14, "etudiant\_7" : 16 , "etudiant\_8" :12 ,"etudiant\_9" :13, "etudiant\_10" : 15}

Écrire un programme qui permet de partager le dictionnaire en trois sous dictionnaires :

* EtudiantsAdmis dont les clés sont les étudiants\_admis et les valeurs des clés sont les moyennes obtenues (moyenne supérieurs ou égales à 10).
* EtudiantsSessionnaire dont les clés sont les etudiants\_session et les valeurs des clés sont les moyennes obtenues entre 8 et 1O (10 non inclue)
* EtudiantsAjournés dont les clés sont les etudiants\_ajournes et les valeurs des clés sont les moyennes obtenues (moyenne inférieure à 8).