

TD1 rappels de python

Ex1 - autre pyramide

boucle, range

Soit l'affichage d'une pyramide vu en cours :

```
for i in range(1,10):
    for j in range(1,10 - i): print(" ",end="")
    for j in range(0,i): print(i,end="")
    for j in range(0,i-1): print(i,end="")
    print()
```

Ecrire le code pour afficher la pyramide suivante :

```
      1
     121
    12321
   1234321
  123454321
 12345654321
1234567654321
123456787654321
12345678987654321
```

Ex2 - moyennes

boucle, liste

1. Écrire un programme qui demande à l'utilisateur d'entrer une liste de N valeurs (N fourni par l'utilisateur) et en calcule la moyenne. Le dialogue utilisateur sera de la forme:: entrer la valeur 1 :... entrer la valeur 2 : ...
2. Calculer la moyenne centrée (on retire du calcul les deux valeurs extrêmes)

Ex3 - températures

alternative, boucle, liste

Écrire un programme qui lit une série de 12 températures. Ce programme affichera la température maximale, la minimale ainsi que la plus proche de 0. Si deux températures sont à la même distance de 0 on retiendra la température négative.

Ex4 - diviseurs

alternative, boucle, liste

Ecrire un programme qui affiche la liste de tous les diviseurs communs entre deux entiers donnés par l'utilisateur.

Ex5- Etude de texte simple

alternative, chaînes de caractère, boucle

Ecrire un programme qui demande à l'utilisateur d'entrer un texte. Ce programme devra afficher : * le nombre de caractères * le nombre de lettres (a-z A-Z) * le nombre de voyelles * le nombre de chiffres

Ex5- Classer les étudiants

dictionnaire, boucle

On considère le dictionnaire de l'exemple ci-dessous dont * les clés sont les noms des élèves * les valeurs des clés sont les résultats obtenus à l'UV INF2

```
INF2 = {"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 , "etudiant_11" : 14 , "etudiant_112" : 9 ,
        "etudiant_13" : 10 , "etudiant_14" : 12 , "etudiant_15" : 13 , "etudiant_16" : 7 ,
        "etudiant_17" : 12 , "etudiant_18" : 15 , "etudiant_19" : 9 , "etudiant_20" : 17}
```

Ecrire un programme Python qui partitionne ce dictionnaire en deux sous dictionnaires : * `etudiantAdmis` dont les clés sont les étudiants admis et les valeurs sont les moyennes obtenues (moyenne supérieure ou égale à 10). * `etudiantNonAdmis` dont les clés sont les étudiants non admis et les valeurs sont les moyennes obtenues (moyenne inférieure à 10).

Le programme affichera les 2 dictionnaires et leur moyenne respective.

Ex6- Création de dictionnaire avec clé de type tuple

chaines de caractère, dictionnaire, boucle enumerate

Soit une liste de noms :

```
noms = ['boris', 'jean', 'simon', 'anna', 'jeanne', 'brandon',
        'theo', 'brahim', 'marion', 'leila', 'lancelot', 'quentin',
        'kevin', 'john', 'louise', 'paul', 'sarah', 'christophe',
        'marielle', 'oriane', 'luc']
```

1. Créer un dictionnaire à partir d'une liste de noms tel que :

* la clé soit un couple (position, lettre) correspondant à la position de la lettre dans le nom * la valeur soit une liste qui contienne tous les noms ayant la lettre à la position donnée

2. Demander à l'utilisateur un couple lettre position et affiche les noms ayant la lettre donnée à la position spécifiée.

Ex6- suite de l'étude de texte

chaines de caractère, dictionnaire, tri

1. Dans la suite de l'exercice 5, calculer la fréquence d'apparition de chaque lettre et afficher les fréquences par ordre décroissant.
2. Afficher le nombre de mots et la taille moyenne des mots

Exemple :

```
Entrez un texte: ca va
Fréquence d'apparition de  a = 50.0 %
Fréquence d'apparition de  c = 25.0 %
Fréquence d'apparition de  v = 25.0 %
Fréquence d'apparition de  b = 0.0 %
Fréquence d'apparition de  d = 0.0 %
...
Fréquence d'apparition de  z = 0.0 %
Nombre de mots : 2
Longueur moyenne des mots : 2
```