GROUPE ISI	Institut Supérieur Informatique	
	Structures de Contrôles	Prof. M. Sy Periode: 2019 – 2020
	Classe(s): Master 1 Data Science & IA($DSIA$)	Durée: –

Exercicie 1

Ecrire un programme demande á l'utilisateur de saisir 3 notes au clavier, ainsi que leurs coefficients, puis affiche ´ á la console la moyenne pondérée, en précisant ajourné si la moyenne est inférieure strictement á 10, et admissible dans le cas contraire. Préciser les limites/lacunes du programme construit.

Exercicie 2

Écrire un programme qui demande á l'utilisateur de saisir au clavier trois nombres réels (i.e. des expressions de type float) a non nul, b et c et qui affiche á la console le nombre de solutions de l'équation

$$ax^2 + bx + c = 0$$

d'inconnue $x \in \mathbb{R}$, ainsi que ses ´éventuelles solutions.

Exercicie 3

creer un objet *vec* qui est liste de lsite $vec = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$

- Creer une liste nommé *flatten* á partir de la liste *vec*

  $flatten = [1, 2, 3, 4, 5, 6, 7, 8, 9]$

Exercicie 4

Cet exercice propose des problèmes assez simples de réduction et de transformation, sur une thématique statistique.

1. Donner une définition de la fonction *somme* qui, étant donné une liste de nombres, renvoie la somme des éléments de cette liste, ou 0 si la liste est vide.
2. Donner une définition de la fonction *moyenne* qui, étant donné une liste non vide de nombres,renvoie la moyenne des éléments de cette liste.
3. Donner une définition de la fonction *carres* qui, étant donné une liste L de nombres, renvoie la liste des carrés des éléments de L.
4. La variance d'une liste de nombres est égale à la différence entre la moyenne des carrés des éléments de la liste et le carré de la moyenne des éléments de la liste. Donner une définition de la fonction *variance* qui, étant donné une liste non vide de nombres, renvoie la *variance* de la liste.
5. L'écart-type d'une liste de nombres est égal à la racine carrée de la variance de la liste. Donner une définition de la fonction *ecart_ type* qui, étant donné une liste non vide de nombres, renvoie l'écart-type de la liste.

Exercicie 5:(Liste obtenues par multiplication ou division)

Dans cet exercice, on résout des problèmes de transformation et de filtrage, ainsi que de combinaison de listes.

- Donner une définition de la fonction *liste_ mult* qui, étant donné une liste L d'entiers et un entier k, retourne la liste obtenue en multipliant par k tous les éléments de L.
- Donner une définition de la fonction *liste_ div* qui, étant donné une liste L d'entiers et un entier k non nul, retourne la liste obtenue en divisant par k les éléments de L qui sont multiples de k et en supprimant les autres.

Exercicie 6: median d'une liste

1. Écrire la fonction $grands(L, x)$ qui reçoit en paramètres une liste de nombres L, et un élément x de L. La fonction renvoie le nombre d'éléments de L qui sont supérieurs strictement à x.
2. Déterminer la complexité de la fonction $grands(L, x)$, et justifier votre réponse.
3. Écrire la fonction $petits(L, x)$ qui reçoit en paramètres une liste de nombres L, et un élément x de L. La fonction renvoie le nombre d'éléments de L qui sont inférieurs strictement à x.

   L est une liste de taille n qui contient des nombres, et m un élément de L. L'élément m est un médian de L, si les deux conditions suivantes sont vérifiées :

   - Le nombre d'éléments de L,qui sont supérieurs strictement à m, est inférieur ou égale à $\frac{n}{2}$
   - Le nombre d'éléments de L, qui sont inférieurs strictement à m, est inférieur ou égale à $\frac{n}{2}$

   Exemple: On considère la liste $L = [25, 12, 6, 17, 3, 10, 20, 12, 15, 38]$, de taille $n = 10$. L'élément 12 est un médian de L, car :
4. Écrire la fonction $median(L)$ qui reçoit en paramètre une liste de nombres L non vide, et qui renvoie un élément médian de la liste L.
5. Déterminer la complexité de la fonction $median(L)$,et justifier votre réponse.

Exercicie 7: Anagrammes

Une anagramme d'un mot M est un mot formé en changeant de place les lettres du mot M.
Exemple :

- 'maire' est une anagramme de 'chien'
- 'niche' est une anagramme de 'chien'
- 'nacre' est une anagramme de 'ancre'

1. Écrire une fonction $Anagramme(C1, C2)$ qui renvoie *True* si la chaîne de caractères $C2$ est un anagramme de $C1$, *False* sinon
2. Ecrire une fonction $listeAnagramme(C1, L)$ qui renvoie *True* si tous les mots de la liste L sont des anagrammes de $C1$, False sinon

   Exemple : si $C1 =' aimer'$ et $L = [' maire',' marie',' ramie',' riame',' mirae']$ alors $listeAnagramme(C1, L)$ renvoie *True*
3. On dispose d'une liste de chaines de caractères L. Ecrire une fonction $NbCar(L)$ qui renvoie une liste de tuples formés de chaque mot de L et du nombre de ses caractères

Exercicie 8: Premières énumérations

1. Définir une fonction baptisée entiers prenant comme arguments deux entiers i et j (avec $i \leq j$) et qui affiche sur une même ligne les entiers de l'intervalle $[|i, j|]$ séparés par le caractère .
2. Modifier votre fonction pour qu'elle affiche désormais les entiers de l'intervalle $[|i, j|]$ qui ne sont pas des multiples de 7.

```
In [1]: entiers(7, 21)
7-8-9-10-11-12-13-14-15-16-17-18-19-20-21

In [2]: entiers(7, 21)
8-9-10-11-12-13-15-16-17-18-19-20
```

Exercicie 10: Graphisme en console

1. Définir une fonction *triangle*1 à un argument entier n qui dessine dans le shell un triangle sur n lignes.
2. Définir une fonction *triangle*2 qui dessine ce même triangle mais dans l'autre sens.
3. Définir une fonction *pyramide*1 qui dessine une pyramide sur $2n1$ lignes.
4. Définir une fonction *pyramide*2 qui dessine une pyramide sur n lignes.

Un exemple pour $n = 5$ de chacune de ces quatre fonctions est présenté figure 2.

```
In [1]: triangle1(5)
*
**
***
****
*****
```

```
In [2]: triangle2(5)
*****
****
***
**
*
```

```
In [3]: pyramide1(5)
*
**
***
****
*****
****
***
**
*
```

```
In [4]: pyramide2(5)
    *
   * *
  * * *
 * * * *
* * * * *
```