

	Institut Supérieur Informatique	
	Structures de Contrôles	PROF. M. SY Periode: 2020 – 2021
	Classe(s): Licence 2 Génie Logiciel (GL)	Durée: –

### Exercice 1

Écrire un programme qui demande à l'utilisateur de saisir 3 notes au clavier, ainsi que leurs coefficients, puis affiche à la console la moyenne pondérée, en précisant ajourné si la moyenne est inférieure strictement à 10, et admissible dans le cas contraire.

### Exercice 2

Écrire un programme qui demande à 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 à 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.

### Exercice 3

Ce jeu est très simple. L'ordinateur tire un nombre au hasard entre 1 et 30 et vous avez cinq essais pour le trouver. Après chaque tentative, l'ordinateur vous dira si le nombre que vous avez proposé est trop grand, trop petit, ou si vous avez trouvé le bon nombre.

Pesent a importer le module random

### Exercice 4: 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 'marie'
- 'niche' est une anagramme de 'chien'
- 'nacre' est une anagramme de 'ancré'

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  $C1 = \text{'aimer'}$  et  $L = [\text{'maire'}, \text{'marie'}, \text{'ramie'}, \text{'riame'}, \text{'mirae'}]$  alors *listeAnagramme*( $C1, L$ ) renvoie *True*

3. On dispose d'une liste de chaînes 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

### Exercice 5: Graphisme en console

1. Définir une fonction *triangle1* à un argument entier  $n$  qui dessine dans le shell un triangle sur  $n$  lignes.
2. Définir une fonction *triangle2* qui dessine ce même triangle mais dans l'autre sens.
3. Définir une fonction *pyramide1* qui dessine une pyramide sur  $2n - 1$  lignes.
4. Définir une fonction *pyramide2* qui dessine une pyramide sur  $n$  lignes.

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

