



**EXAMEN SEMESTRE1: INFORMATIQUE**

Durée : 2h

Filières : MP1, PC1 et PT1

Nombre de pages : 4

**Exercice 1 (6 points)**

Soit  $L1$  une liste de taille  $N$  ( $0 < N \leq 50$ ) dont les éléments sont des entiers non nuls triés **dans l'ordre décroissant** et  $L2$  une liste de la même taille  $N$  contenant des éléments dont la valeur ne peut être que 0 ou 1.

- ✓ La compression de  $L1$  par  $L2$  donne une liste  $C$ .
- ✓ La liste  $C$  est remplie par les éléments de la liste  $L1$ . Un élément  $i$  de la liste  $L1$  est ajouté à la liste  $C$  à condition que  $L2[i]=1$ .
- ✓ Les éléments apparaissent dans le même ordre dans les deux listes  $L1$  et  $C$  (voir l'exemple).

**Exemple :**

Soient  $L1$  et  $L2$  deux listes initiales :

$L1=[89,47,25,9,4,2,-2]$

$L2=[0,1,1,0,1,0,1]$

La liste  $C$  résultat de compression de la liste  $L1$  par  $L2$  est la suivante :

$C=[47,25,4,-2]$

Ecrire un programme python permettant de :

1. Saisir  $N$ , la liste  $L1$  et la liste  $L2$  (faire le contrôle de saisie nécessaire).
2. Remplir la liste  $C$ .
3. Afficher par ordre croissant la liste  $C$ .

**Exercice 2 (8 points)**

Dans cet exercice on cherche à évaluer la force des mots de passe des différents utilisateurs d'un site web.

Un mot de passe est une chaîne de caractères qui ne comporte pas d'espaces.

La force d'un mot de passe varie, selon la valeur d'un **Score** calculé, de 'Très faible' jusqu'à 'Très fort' :

- Si le score  $< 20$ , la force du mot de passe est '**Très faible**'
- Sinon si le score  $< 40$ , la force d'un mot de passe est '**Faible**'
- Sinon si le score  $< 80$ , la force du mot de passe est '**Fort**'
- Sinon la force du mot de passe est '**Très Fort**'

Le score se calcule en additionnant des bonus et en retranchant des pénalités.

Les bonus attribués sont :

- Nombre total de caractères \* 4
- (Nombre total de caractères – nombre de lettres majuscules) \* 2
- (Nombre total de caractères – nombre de lettres minuscules) \* 3
- Nombre de caractères non alphabétiques \* 5

Les pénalités imposées sont :

- La longueur de la plus longue séquence de lettres minuscules \* 2
- La longueur de la plus longue séquence de lettres majuscules \* 3

**Exemple :**

Pour le mot de passe 'P@cSI\_promo2017', le score se calcule comme suit :

La somme de bonus =  $15 * 4 + (15 - 3) * 2 + (15 - 6) * 3 + 6 * 5 = 141$

- Le nombre total de caractères = 15
- Le nombre de lettres majuscules = 3
- Le nombre de lettres minuscules = 6
- Le nombre de caractères non alphabétiques = 6

La somme des pénalités =  $5 * 2 + 2 * 3 = 16$

- La longueur de la plus longue séquence de lettres minuscules ('promo') = 5
- La longueur de la plus longue séquence de lettres majuscules ('SI') = 2

Le score final =  $141 - 16 = 125$  ; puisque  $125 > 80$  alors le mot de passe est considéré comme 'Très fort'

### Travail demandé :

Ecrire un programme python qui permet de :

1. Saisir une chaîne correspondante au mot de passe
2. Vérifier si la chaîne ne contient pas des espaces, si la chaîne contient des espaces on doit les supprimer.
3. Calculer le nombre de caractères minuscules **Nb\_min**.
4. Calculer le nombre de caractères majuscules **Nb\_max**.
5. Calculer le nombre de caractères non alphabétiques **Nb\_NonAlph**.
6. Calculer longueur de la plus longue séquence de lettres majuscules **Long\_Maj**.
7. Calculer la longueur de la plus longue séquence de lettres minuscules **Long\_Min**.
8. Calculer la somme des bonus, la somme des pénalités et le score de mot de passe sachant que :  
le score = la somme des bonus – la somme des pénalités.
9. Afficher la force du mot de passe saisi selon la valeur du score.

### Exercice 3 (6 points)

Dans cet exercice, une famille est représentée par un dictionnaire dont les éléments sont des associations entre une clé qui est le prénom d'une personne et une valeur qui est une liste des prénoms des enfants de cette personne.

Si une personne n'a pas d'enfants alors elle n'apparaît pas comme clé du dictionnaire.

Dans tout l'exercice on fait l'hypothèse que les personnes ont toutes des prénoms différents.

Ecrire un script python qui permet de :

1. Créer le dictionnaire `D_Famille = {'eve': ['tom', 'lea', 'luc', 'kim'], 'tom': ['isa', 'bob'], 'lea': ['ali'], 'bob': ['bea', 'tim'], 'luc': ['sam', 'jon', 'lou'], 'sam': ['ana', 'guy'], 'guy': ['ben']}`

Dans cet exemple :



- tom, lea, luc et kim sont les enfants d'eve
- isa et bob sont les enfants de tom (et donc les petits-enfants d'eve)
- lou n'est pas une clé du dictionnaire : lou n'a pas d'enfants

2. Saisir une chaîne de caractères  $p$ , puis afficher les prénoms des enfants de  $p$  (si elle existe dans le dictionnaire). Le message «  $p$  n'est pas une clé du dictionnaire » sera affiché dans le cas où  $p$  n'a pas d'enfants.

**Exemple d'affichage :**

*L'ensemble des enfants de 'eve' sont 'kim', 'lea', 'tom' et 'luc'*

3. Saisir une chaîne de caractères  $x$ , puis afficher les petits enfants de  $x$  (si  $x$  a des petits d'enfants). Le message «  $x$  n'a pas de petits-enfants » sera affiché dans le cas où  $x$  n'a pas de petits-enfants.

**Remarque :** les petits-enfants de  $x$  sont les enfants des enfants de  $x$ .

**Exemple d'Affichage :**

*Les petits enfants de 'eve' sont 'lou', 'isa', 'bob', 'ali', 'jon' et 'sam'*

4. Saisir une chaîne de caractères  $y$ , puis afficher les parents de  $y$  ( $p$  est un parent de  $y$  si  $y$  est un enfant de  $p$ ). Si  $y$  n'a pas de parent on doit afficher le message «  $y$  n'a pas de parent ».

**Exemple d'affichage :**

*Le parent de tom est eve.*

5. Soit l'ensemble  $X = \{\text{'tom'}, \text{'lea'}\}$  afficher l'ensemble des prénoms de leurs enfants.

**Exemple d'affichage :**

*Les enfants de {'tom', 'lea'} sont : {'isa', 'bob', 'ali'}*