

Important :

Le candidat est appelé à créer, dans le répertoire **BAC2025**, un dossier de travail ayant comme nom son numéro d'inscription (6 chiffres) et dans lequel il doit enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.

Mots ordonnés

On se propose de concevoir une interface graphique permettant de saisir une chaîne de caractères **Ch** et d'afficher sans redondance ses mots qui sont triés selon l'ordre alphabétique, séparés par le caractère "-". Un mot est trié selon l'ordre alphabétique si tous ses caractères sont ordonnés en ordre croissant des lettres alphabétiques.

Exemples :

- Pour **Ch="le lion mange la viande."**, le programme affiche : **Aucun mot trié dans la chaine**
- Pour **Ch="il est gentil."**, le programme affiche : **il-est**
En effet, les caractères des deux mots "il" et "est" sont triés en ordre croissant alphabétiquement.
- Pour **Ch="il est fou."**, le programme affiche : **il-est-fou**
En effet, les caractères des trois mots "il", "est" et "fou" sont triés en ordre croissant alphabétiquement.
- Pour **Ch="il est gentil et il est fou."** le programme affiche : **il-est-et-fou**
En effet, les caractères des mots "il", "est", "et" et "fou" sont triés en ordre croissant alphabétiquement et les mots "il" et "est" sont affichés une seul fois chacun.

Travail demandé

- 1) Créer l'interface graphique illustrée dans la figure **Fig-1** et l'enregistrer sous le nom **Interface**.

Cette interface contient les éléments suivants :

- Un label contenant le texte : "**Mots ordonnés**".
- Un label contenant le texte : "**La chaine :**".
- Une zone de saisie pour la saisie de la chaîne de caractères **Ch**.
- Un bouton intitulé "**Afficher**".
- Un label dédié à l'affichage.

Mots ordonnés

La chaine: []

Afficher

Fig-1

- 2) Créer un programme python et l'enregistrer sous le nom **MotOrd**, dans lequel on demande :
- d'implémenter l'algorithme suivant de la fonction **Ordonner (Mot)** qui permet de vérifier si la chaîne de caractères **Mot** est triée en ordre croissant des lettres alphabétique ou non.

Fonction **Ordonner (Mot : Chaîne de caractères) : Booléen**

DEBUT

$i \leftarrow 1$

Tant que $i < \text{Long}(\text{Mot})$ **Et** ($\text{Mot}[i-1] \leq \text{Mot}[i]$) **Faire**
 $i \leftarrow i+1$

Fin Tant que

Retourner $i = \text{Long}(\text{Mot})$

FIN

| T.D.O.L | |
|----------------|--------------------|
| Objet | Type/Nature |
| i | Entier |

- de développer une fonction **Traitemet (ch)** qui permet d'exploiter la fonction **Ordonner** et de retourner les mots de la chaine de caractères **ch** qui sont ordonnés sans redondance séparés par le caractère "-".
- de développer un module **Play**, qui s'exécute suite au clic sur le bouton "**Afficher**", permettant :
 - de récupérer la chaine de caractères saisie, qui doit être une chaine de caractères non vide, se termine par le caractère ".", formée au maximum par 30 caractères (lettres minuscules et espaces uniquement) et deux mots consécutifs sont séparés par un seul espace. Si au moins une contrainte n'est pas respectée, afficher le message "*Veuillez saisir une chaine valide !*" comme illustré dans les figures **Fig-2** et **Fig-3**.
 - d'exploiter la fonction **Traitemet** afin d'afficher sans redondance les mots triés via le label dédié à l'affichage, comme illustré dans les figures **Fig-4**, **Fig-5**, **Fig-6** et **Fig-7**.
- d'exploiter l'annexe présentée ci-après tout en apportant les modifications nécessaires à l'intégration de l'interface graphique **Interface**.

Mots ordonnés

La chaine:

Afficher

Veuillez saisir une chaine valide !

Fig-2

Mots ordonnés

La chaine:

Afficher

Veuillez saisir une chaine valide !

Fig-3

Mots ordonnés

La chaine:

Afficher

il-est

Fig-4

Mots ordonnés

La chaine:

Afficher

il-est-fou

Fig-5

Mots ordonnés

La chaine:

Afficher

il-est-et-fou

Fig-6

Mots ordonnés

La chaine:

Afficher

Aucun mot trié dans la chaine

Fig-7

Annexe

```
from PyQt5.uic import loadUi  
from PyQt5.QtWidgets import QApplication  
.....  
.....  
app = QApplication([])  
windows = loadUi ("Nom_Interface.ui")  
windows.show()  
windows.Nom_Bouton.clicked.connect (Nom_Module)  
app.exec_()
```

Grille d'évaluation

| Tâches | Nombre de points |
|---|------------------|
| 1) Création de l'interface Interface . | 3 |
| 2) Création du programme MotOrd | 17 = |
| a) Implémentation de la fonction Ordonner | 4 |
| b) Développement de la fonction Traitement | 5.5 |
| c) Développement du module Play | 6 |
| d) Exploitation de l'annexe. | 1.5 |