

RÉPUBLIQUE TUNISIENNE *** Ministère de l'Éducation	EXAMEN DU BACCALAURÉAT	SESSION 2023
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Sections : Mathématiques, Sciences expérimentales et Sciences techniques	
	Coefficient de l'épreuve : 0.5	Durée : 1h

**Important :**

1. Une solution modulaire au problème posé est exigée.
2. Dans le répertoire **Bac2023**, créez un dossier de travail ayant comme nom votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solution à ce sujet.

## Miroirs de mots

On se propose de concevoir une interface graphique permettant de saisir une chaîne de caractères **ch** et de la crypter en formant une nouvelle chaîne par les miroirs des mots de **ch** dans leur ordre d'apparition. On rappelle que le miroir d'un mot consiste à permuter le premier caractère avec le dernier, le deuxième caractère avec l'avant dernier et ainsi de suite.

**Exemple :**

Pour **ch** = "La vie est une aventure merveilleuse"

On obtient la chaîne cryptée : "aL eiv tse enu erutneva esuellievrem"

L'interface graphique à concevoir contient les éléments suivants, comme l'illustre la capture d'écran ci-dessous :

- Un label contenant le texte "**Miroirs de mots**"
- Un label contenant le texte "**Introduire une chaîne :** "
- Une zone de saisie pour la saisie d'une chaîne
- Un label pour afficher le résultat
- Un bouton intitulé "**Miroir**"

**Travail demandé :**

- 1) Concevoir l'interface graphique présentée précédemment et l'enregistrer sous le nom **InterfaceMiroirsMots**.
- 2) Créer un programme Python et l'enregistrer sous le nom **MiroirsMots**, dans lequel, il est demandé :
  - a) de développer une fonction nommée **Miroir (M)** qui permet de retourner le miroir d'un mot **M**.
  - b) de développer un module **Play**, qui s'exécute suite à un clic sur le bouton "**Miroir**", permettant :
    - de récupérer la chaîne **ch** saisie. La chaîne **ch** doit être non vide et de longueur inférieure à 50, contient seulement des lettres alphabétiques en minuscule et chaque deux mots consécutifs sont séparés par un seul espace.
    - de déterminer la chaîne cryptée en utilisant la fonction **Miroir (M)** et d'afficher le résultat via le **label** dédié à l'affichage dans l'interface graphique **InterfaceMiroirsMots**.
  - c) d'ajouter les instructions permettant d'exploiter l'interface graphique intitulée **InterfaceMiroirsMots** en se référant à l'annexe ci-après.

*N.B. : l'affichage doit être conforme aux exemples d'exécutions suivants :*

**Exemples d'exécutions :**

**Miroirs de mots**

Introduire une chaîne :

Veuillez introduire une chaîne

**Miroirs de mots**

Introduire une chaîne :

Entre 2 mots un seul espace est autorisé

**Miroirs de mots**

Introduire une chaîne :

aL eiv tse enu erutneva esuellievrem

```
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
Conception de l'interface " <b>InterfaceMiroirsMots</b> "	<b>4 pts</b>
Création et enregistrement du programme " <b>MiroirsMots</b> "	<b>1 pt</b>
Développement de la fonction " <b>Miroir</b> "	<b>4 pts</b>
Développement du module " <b>Play</b> "	<b>6 pts</b>
Ajout des instructions de l'exploitation de l'interface	<b>3 pts</b>
Modularité et cohérence	<b>2 pts</b>