

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.

## Rotation - Miroir

On se propose de concevoir une interface graphique permettant de saisir une chaîne de caractères **ch** et de la crypter en utilisant le principe suivant :

➤ **Étape 1 : Rotation13**

Effectuer, pour chaque caractère de **ch**, une rotation de **13** caractères dans l'ordre alphabétique en appliquant la formule  $\text{Chr}(97 + (\text{Ord}(\text{ch}[i]) - 97 + 13) \text{ Mod } 26)$  pour obtenir le caractère correspondant.

➤ **Étape 2 : Miroir**

Transformer la chaîne de caractères obtenue à l'issue de l'**Étape 1** en son miroir. C'est-à-dire permuter le premier caractère avec le dernier, le deuxième caractère avec l'avant dernier et ainsi de suite.

**Exemple :**

Pour **ch** = "algo"

➤ **Étape 1 : Rotation13**

- le caractère "a" sera crypté en "n". En effet,  $\text{Chr}(97+(\text{Ord}(\text{"a"})-97+13) \text{ Mod } 26) = \text{Chr}(110)$  qui correspond au caractère "n"
- le caractère "l" sera crypté en "y". En effet,  $\text{Chr}(97+(\text{Ord}(\text{"l"})-97+13) \text{ Mod } 26) = \text{Chr}(121)$  qui correspond au caractère "y".
- le caractère "g" sera crypté en "t". En effet,  $\text{Chr}(97+(\text{Ord}(\text{"g"})-97+13) \text{ Mod } 26) = \text{Chr}(116)$  qui correspond au caractère "t".
- le caractère "o" sera crypté en "b". En effet,  $\text{Chr}(97+(\text{Ord}(\text{"o"})-97+13) \text{ Mod } 26) = \text{Chr}(98)$  qui correspond au caractère "b".

D'où le résultat de l'**étape 1** est "nytb"

➤ **Étape 2 : Miroir**

Le miroir de la chaîne "nytb" est "btyn"

Donc, le cryptage de la chaîne "algo" en appliquant le principe ci-dessus est "btyn"

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

- Un label contenant le texte "**Rotation - Miroir**"
- 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 "**Crypter**"

**Rotation - Miroir**

Introduire une chaîne :

**Travail demandé :**

- 1) Concevoir l'interface graphique présentée précédemment et l'enregistrer sous le nom **InterfaceRotationMiroir**.
- 2) Créer un programme Python et l'enregistrer sous le nom **RotationMiroir**, dans lequel, il est demandé :
  - a) d'implémenter l'algorithme suivant de la fonction **Rotation** qui permet de retourner le résultat de l'étape 1.

**Fonction** Rotation (ch : Chaîne de caractères) : **Chaîne de caractères**  
**DEBUT**  
 res ← ""  
**Pour** i de 0 à Long(ch)-1 **Faire**  
   res ← res + Chr(97 + (Ord(ch[i]) - 97 + 13) Mod 26)  
**Fin Pour**  
**Retourner** (res)  
**FIN**

**Déclaration des objets**

Objet	Type/Nature
i	Entier
res	Chaîne de caractères

- b) de développer une fonction nommée **Miroir (ch)** qui permet de retourner le miroir d'une chaîne de caractères **ch**.
- c) de développer un module **Play**, qui s'exécute suite à un clic sur le bouton "**Crypter**", permettant :
  - de récupérer la chaîne **ch** saisie. La chaîne **ch** doit être non vide, de longueur inférieure à 10 et contient seulement des lettres alphabétiques en minuscule.
  - de déterminer la chaîne cryptée en utilisant les deux fonctions **Rotation (ch)** et **Miroir (ch)** et d'afficher le résultat via le **label** dédié à l'affichage dans l'interface graphique **InterfaceRotationMiroir**.
- d) d'ajouter les instructions permettant d'exploiter l'interface graphique intitulée **InterfaceRotationMiroir** en se référant à l'annexe ci-après.

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

**Exemples d'exécutions :**

**Rotation - Miroir**

Introduire une chaîne :

Veuillez introduire une chaîne

**Crypter**

**Rotation - Miroir**

Introduire une chaîne :

Veuillez introduire une chaîne valide !

**Crypter**

**Rotation - Miroir**

Introduire une chaîne :

La chaîne cryptée est : btyn

**Crypter**

**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**

<b>Tâches</b>	<b>Nombre de points</b>
Conception de l'interface " <b>InterfaceRotationMiroir</b> "	<b>4 pts</b>
Création et enregistrement du programme " <b>RotationMiroir</b> "	<b>1 pt</b>
Implémentation de la fonction " <b>Rotation</b> "	<b>3 pts</b>
Développement de la fonction " <b>Miroir</b> "	<b>3 pts</b>
Développement du module " <b>Play</b> "	<b>4 pts</b>
Ajout des instructions de l'exploitation de l'interface	<b>3 pts</b>
Modularité et cohérence	<b>2 pts</b>