

|                                                            |                                                                             |            |
|------------------------------------------------------------|-----------------------------------------------------------------------------|------------|
| RÉPUBLIQUE TUNISIENNE<br>-----<br>MINISTÈRE DE L'ÉDUCATION | EXAMEN DU BACCALAURÉAT<br>SESSION 2026                                      |            |
|                                                            | ÉPREUVE PRATIQUE D'INFORMATIQUE                                             |            |
|                                                            | Sections : Mathématiques, Sciences expérimentales et<br>Sciences techniques |            |
|                                                            | Coefficient de l'épreuve : 0.5                                              | Durée : 1h |

### Important

Le candidat est appelé à créer dans le répertoire **Bac2026**, 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.

## Clé de cryptage

L'**American Standard Code for Information Interchange (ASCII)** est un système de codage qui associe à chaque caractère un nombre représenté en mémoire sur **8 bits**.

Un caractère imprimable, ayant un code ASCII entre **32** et **126**, est un symbole (lettre, chiffre, signe de ponctuation, etc.) destiné à être affiché sur l'écran ou imprimé.

**Exemple** : Le code ASCII du caractère "**A**" est **65** et sa représentation binaire est **01000001**

**NB.** : Le caractère espace est un caractère imprimable de code ASCII **32**.

Afin d'assurer la confidentialité des échanges des données, le responsable informatique d'une entreprise utilise une clé de cryptage pour chiffrer les données avant une éventuelle transmission.

Dans ce contexte, on envisage de concevoir une application, enrichie par une interface graphique, qui simule la création et l'affichage de cette clé de cryptage.

Cette application permet de :

- saisir un entier **N** multiple de 8 (avec  $32 \leq N \leq 64$ ),
- remplir un tableau **T** par **N** valeurs binaires (**0** ou **1**) générées aléatoirement,
- coder chaque séquence de **8** valeurs binaires du tableau **T** en un caractère imprimable à l'aide de la fonction **Coder** définie ci-après (page 2 sur 3),
- former une clé de cryptage en concaténant, dans leur ordre d'apparition, tous les caractères imprimables obtenus.

### Exemple :

Pour **N = 32** et le tableau **T** suivant :

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  | 1  | 1  | 0  | 0  |    |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |

Le programme affiche : La clé est **B?r\***

En effet,

- ✓ La concaténation des 8 bits de la première séquence de **T** est **01000010**, qui correspond au caractère imprimable "**B**".
- ✓ La concaténation des 8 bits de la deuxième séquence de **T** est **00111111**, qui correspond au caractère imprimable "**?**".
- ✓ La concaténation des 8 bits de la troisième séquence de **T** est **11100001**, son codage en caractère imprimable est "**r**".
- ✓ La concaténation des 8 bits de la quatrième séquence de **T** est **00011100**, son codage en un caractère imprimable est "**\***".

## Travail demandé

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

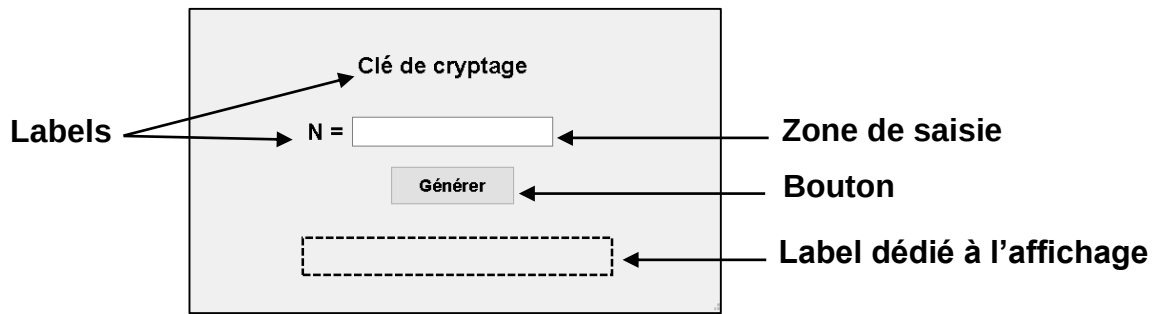


Fig-1

2. Créer un programme en python et l'enregistrer sous le nom **Clé**, dans lequel on demande :
  - a. D'écrire un module **Remplir (T, N)** qui permet de charger aléatoirement un tableau **T** par **N** chiffres binaires (**0** ou **1**).
  - b. D'implémenter l'algorithme suivant de la fonction **Coder (CH)** permettant de transformer une chaîne de caractères de **8** bits en un caractère imprimable.

**Fonction Coder (CH : Chaîne de caractères) : Caractère**

**DEBUT**

$x \leftarrow 0$

$p \leftarrow 1$

**Pour**  $i$  de  $\text{long}(\text{CH}) - 1$  à  $0$  ( $\text{pas} = -1$ ) **Faire**

**Si**  $\text{CH}[i] = "1"$  **Alors**

$x \leftarrow x + p$

**FinSi**

$p \leftarrow p * 2$

**Fin Pour**

**Si**  $x < 32$  **Ou**  $x > 126$  **Alors**

$x \leftarrow 32 + x * 94 \text{ Div } 255$

**FinSi**

**Retourner**  $\text{Chr}(x)$

**FIN**

| T.D.O.L   |             |
|-----------|-------------|
| Objet     | Type/Nature |
| $i, p, x$ | Entier      |

- c. D'écrire une fonction **Former (T, N)** qui retourne une chaîne de caractères formée par la concaténation des caractères imprimables correspondants au codage de chaque séquence de **8** valeurs binaires du tableau **T**. Ce module doit faire appel à la fonction **Coder**.
- d. D'écrire un module **Play**, qui s'exécute suite au clic sur le bouton "**Générer**", permettant :
  - ✓ de récupérer la valeur du nombre **N** saisi, de s'assurer de sa validité comme décrit précédemment et d'afficher, le cas échéant, le message adéquat via le label dédié à l'affichage, comme illustré dans les figures **Fig-2**, **Fig-3** et **Fig-4**,
  - ✓ d'exploiter les modules **Remplir** et **Former** afin d'afficher le message adéquat via le label dédié à l'affichage, comme illustré dans la figure **Fig-5**,
- e. D'exploiter l'annexe présentée ci-après tout en apportant les modifications nécessaires à l'intégration de l'interface graphique **InterfaceClé**.

Clé de cryptage

N =

Veuillez saisir un nombre

**Fig-2**

Clé de cryptage

N =

N doit être :  $32 \leq N \leq 64$  et multiple de 8

**Fig-3**

Clé de cryptage

N =

N doit être :  $32 \leq N \leq 64$  et multiple de 8

**Fig-4**

Clé de cryptage

N =

La clé est B?r\*

**Fig-5**

| Annexe                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> from PyQt5.uic import loadUi from PyQt5.QtWidgets import QApplication from numpy import * import random T=array([int]*64) ..... ..... app = QApplication([]) windows = loadUi ("Nom_Interface.ui") windows.show() windows.Nom_Bouton.clicked.connect (Nom_Module) app.exec_() </pre> |

### Grille d'évaluation

| Tâches                                           | Nombre de points |
|--------------------------------------------------|------------------|
| 1. Création de l'interface <b>InterfaceClé</b> . | 3                |
| 2. Création du programme <b>Clé</b> .            | 17 =             |
| a. Élaboration du module <b>Remplir</b>          | 3                |
| b. Implémentation de la fonction <b>Coder</b>    | 5                |
| c. Élaboration du module <b>Former</b>           | 4.5              |
| d. Élaboration du module <b>Play</b> .           | 3.5              |
| e. Exploitation de l'annexe                      | 1                |