

RÉPUBLIQUE TUNISIENNE ----- MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT SESSION 2026	
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Sections : Mathématiques, Sciences expérimentales et 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.

Code secret

Un joueur piégé dans une pièce fermée, dispose d'un paquet d'étiquettes de lettres alphabétiques distinctes. Pour sortir, il doit trouver le code secret d'ouverture de la porte qui est représenté par les couples de lettres alphabétiques ayant un écart minimal absolu entre leurs codes ASCII.

Dans ce contexte, on envisage de développer une application, enrichie par une interface graphique, qui simule la détermination de ce code secret selon le principe suivant :

- saisir un entier **N** ($5 \leq N \leq 26$),
- remplir un tableau **T** par **N** lettres alphabétiques majuscules distinctes générées aléatoirement,
- déterminer l'écart minimal absolu entre les codes **ASCII** des différents couples de lettres (**T[i]**, **T[j]**) avec $i < j$,
- déterminer la clé formée par la concaténation des lettres alphabétiques de chaque couple ayant cet écart minimal.

Exemple :

Pour le tableau **T** suivant contenant **5** lettres alphabétiques majuscules distinctes :

T	A	R	H	F	P
	0	1	2	3	4

Les couples sont : (A,R) ; (A,H) ; (A,F) ; (A,P) ; (R,H) ; (R,F) ; (R,P) ; (H,F) ; (H,P) ; (F,P)

Le tableau suivant présente l'ensemble des couples de lettres ainsi que l'écart absolu entre leurs codes ASCII respective :

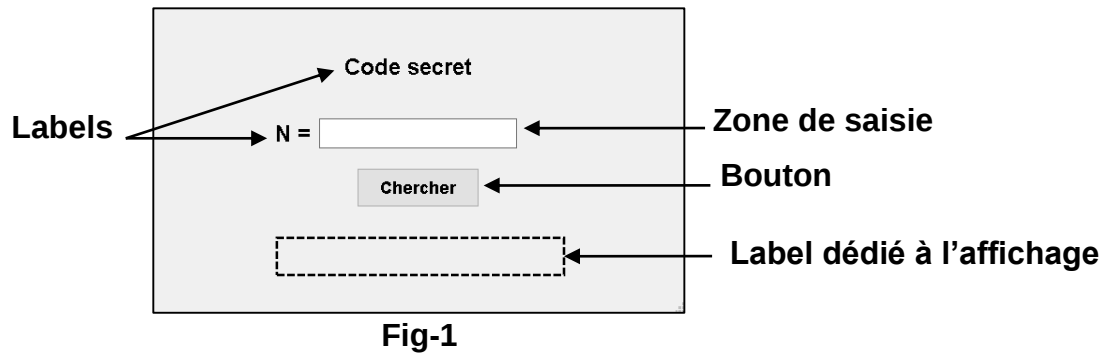
Couple	Ecart Absolu
(A,R)	$ \text{ASCII}("A") - \text{ASCII}("R") = 65-82 = 17$
(A,H)	$ \text{ASCII}("A") - \text{ASCII}("H") = 65-72 = 7$
(A,F)	$ \text{ASCII}("A") - \text{ASCII}("F") = 65-70 = 5$
(A,P)	$ \text{ASCII}("A") - \text{ASCII}("P") = 65-80 = 15$
(R,H)	$ \text{ASCII}("R") - \text{ASCII}("H") = 82-72 = 10$
(R,F)	$ \text{ASCII}("R") - \text{ASCII}("F") = 82-70 = 12$
(R,P)	$ \text{ASCII}("R") - \text{ASCII}("P") = 82-80 = \mathbf{2}$
(H,F)	$ \text{ASCII}("H") - \text{ASCII}("F") = 72-70 = \mathbf{2}$
(H,P)	$ \text{ASCII}("H") - \text{ASCII}("P") = 72-80 = 8$
(F,P)	$ \text{ASCII}("F") - \text{ASCII}("P") = 70-80 = 10$

➔ L'écart minimal absolu est égale à **2**.

➔ Le code secret est **RP-HF**

Travail demandé

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



- Créer un programme en python et l'enregistrer sous le nom **CodeSecret**, dans lequel on demande :
 - D'implémenter l'algorithme suivant de la procédure **Remplir** qui permet de remplir un tableau **T** par **N** lettres alphabétiques majuscules distinctes, générées aléatoirement.

Procédure Remplir (@ T : TAB, N : Entier)									
DEBUT $T[0] \leftarrow \text{Chr}(65 + \text{Aléa}(0, 25))$ Pour i de 1 à N-1 Faire Répéter $T[i] \leftarrow \text{Chr}(65 + \text{Aléa}(0, 25))$ Jusqu'à Existe (T, i-1, T[i]) = Faux Fin Pour FIN	<table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 10px;"> <thead> <tr> <th style="text-align: center; padding: 2px 5px;">Nouveau type</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; padding: 2px 5px;">TAB = Tableau de 26 Caractère</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 2px 5px;">Objet</th> <th style="text-align: center; padding: 2px 5px;">Type/Nature</th> </tr> </thead> <tbody> <tr> <td style="text-align: center; padding: 2px 5px;">i</td> <td style="text-align: center; padding: 2px 5px;">Entier</td> </tr> <tr> <td style="text-align: center; padding: 2px 5px;">Existe</td> <td style="text-align: center; padding: 2px 5px;">Fonction</td> </tr> </tbody> </table>	Nouveau type	TAB = Tableau de 26 Caractère	Objet	Type/Nature	i	Entier	Existe	Fonction
Nouveau type									
TAB = Tableau de 26 Caractère									
Objet	Type/Nature								
i	Entier								
Existe	Fonction								

- D'écrire une fonction booléenne **Existe** (T, K, C) qui permet de vérifier l'existence d'un caractère **C** dans un tableau **T** contenant **K** caractères.
- D'écrire une fonction **EcartMin** (T, N) qui retourne la valeur absolue de l'écart minimal entre les codes **ASCII** des couples de lettres du tableau **T**.
- D'écrire une fonction **Couple** (T, N) qui détermine une chaîne de caractères formée par la concaténation des lettres des couples du tableau **T**, ayant un écart minimal séparés, par le symbole "-". Cette fonction doit exploiter la fonction **EcartMin**.
- D'écrire un module **Play**, qui s'exécute suite au clic sur le bouton "**Chercher**", 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** et **Fig-3**,
 - ✓ d'exploiter les modules **Remplir** et **Couple** afin d'afficher le message adéquat via le label dédié à l'affichage, comme illustré dans la figure **Fig-4**.
- D'exploiter l'annexe présentée ci-après tout en apportant les modifications nécessaires à l'intégration de l'interface graphique **InterfaceCode**.

Code secret

N =

Veuillez saisir un nombre !

Fig-2

Code secret

N =

N doit être compris entre 5 et 50

Fig-3

Code secret

N =

Le code secret : RP-HF

Fig-4

Annexe

```
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QApplication
from numpy import *
import random
T=array([str]*26)
.....
.....
app = QApplication([])
windows = loadUi ("Nom_Interface.ui")
windows.show()
windows.Nom_Bouton.clicked.connect (Nom_Module)
app.exec_()
```

Grille d'évaluation

Tâche	Nombre de points
1. Création de l'interface InterfaceCode	3
2. Création du programme CodeSecret	17 =
a) Implémentation du module Remplir	3
b) Élaboration du module Existe	1.5
c) Élaboration du module EcartMin	3
d) Élaboration du module Couple	5
e) Élaboration du module Play	3.5
f) Exploitation de l'annexe	1