

## 1 Principe de codage

**Question 1.** Expliquer en quelques lignes et en utilisant une représentation graphique le principe du chiffre de Vigenère.

## 2 Manipuler les fichiers texte

**Objectif :** Apprendre à manipuler des fichiers texte en Python de façon à :

- ouvrir, fermer un fichier
- lire et écrire du texte dans un fichier

### 2.1 Lecture d'un fichier texte

Vous disposez d'un fichier python qui permet de lire un fichier texte.

Nom du fichier : **liretexteEleve2.py**

Programme 2	
fichier="texte.txt"	#Nom du fichier traité
file = open(fichier, "r")	#Ouverture du fichier
for ligne in file:	#Lecture du contenu du fichier
print(ligne)	
file.close()	#Fermeture du fichier
print(type(ligne))	#Affiche le type de variable

**Question 2.** Tester le programme mis à disposition en mode « Déboguer », voir Document Ressource DR1. Analyser l'évolution du contenu des variables notamment la variable « ligne ». Préciser le type de la variable « ligne ».

Votre camarade de projet a testé un programme qui permet aussi de lire un fichier texte mais qui est différent du votre.

**Question 3.** A partir des critères d'analyse définis, **comparer** les deux programmes.

*Pour la suite du projet on utilisera le programme 1.*

### 2.2 Ecriture dans un fichier texte

Vous disposez d'un fichier python qui permet de lire un fichier texte.

Nom du fichier : **ecrituretexteEleve2.py**

Elève 2
fichier="texteEcrire2.txt"
fichier = open(fichier, "a")
fichier.write("nouveau texte 2")
fichier.close()

**Question 4.** Tester le programme mis à disposition. Analyser le résultat en ouvrant le fichier texte modifié.

Votre camarade de projet a testé un programme qui permet aussi d'écrire dans un fichier texte mais qui est différent du votre.

**Question 5.** Comparer vos analyses et choisir le programme que vous conserveriez pour la suite du projet. Justifier votre réponse.

## 3 Format du fichier texte

Le texte contenu dans le fichier texte et qui devra ensuite être codé, ne comportera que des lettres minuscules non accentuées et des espaces et des retours à la ligne (saut de ligne).

## 4 Implémenter le codage

### 4.1 Fichiers texte

**Question 6.** Ecrire une fonction qui permet d'écrire dans un fichier texte :

- Nom de la fonction : **ecrireFichier()**
- Paramètre : **nomFichier** et **texteCode**
- Renvoi : **aucun**

**Question 7.** Tester la fonction afin de vérifier son bon fonctionnement.

### 4.2 Caractères et nettoyage

Le travail fait précédemment nous a permis de récupérer l'ensemble du texte contenu dans un fichier. Il nous faut maintenant le préparer en vue de son codage. Deux étapes sont nécessaires :

- Décomposer la chaîne de caractères contenant le texte de façon à placer chacune des lettres dans une liste (« texte » → ['t', 'e', 'x', 't', 'e'])
- Supprimer les caractères particuliers, ici les espaces ' ' et les sauts de lignes matérialisés par \n ([ 't', 'e', 'x', 't', 'e', ' ', ' ', 'l', 'u', '\n' ] → ['t', 'e', 'x', 't', 'e', 'l', 'u' ])

Pour cela on va créer une **fonction** pour chaque **étape**. Chacune sera écrite depuis un algorithme fournis.

**Question 8.** Ecrire une fonction qui permet de supprimer les caractères particuliers :

- Nom de la fonction : **supCarPar()**
- Paramètre : **listeLettres**
- Renvoi : **listeLettres**
- La liste **listeLettres** est stockée dans une variable de type liste de même nom

Algorithme
<b>DEBUT DE L'ALGORITHME</b> <b>POUR</b> chaque caractère <b>car</b> de la liste contenant toutes les lettres <b>DEBUT POUR</b> Si le caractère est un espace OU si le caractère est un saut de ligne <b>DEBUT SI</b> Supprimer le caractère de la liste <b>FIN SI</b> <b>FIN POUR</b> <b>RETOURNER</b> la liste contenant les lettres après nettoyage <b>FIN DE L'ALGORITHME</b>

**Question 9.** Tester la fonction afin de vérifier son bon fonctionnement.



**Question 10.** Envoyer votre travail via le travail à faire de l'ENT. Le format du fichier est le suivant : **cesarE2-J1.py**

### 4.3 Mise en commun des fonctions

**Question 11.** Regrouper dans un seul fichier les différentes fonctions implémentées, soit :

<ul style="list-style-type: none"> <li>• lireFichier()</li> <li>• extraire()</li> </ul>	<ul style="list-style-type: none"> <li>• écrireFichier()</li> <li>• supCarPar()</li> </ul>
---	--

**Question 12.** Tester le programme afin de vérifier son bon fonctionnement.



**Question 13.** Envoyer votre travail via le travail à faire de l'ENT. Le format du fichier est le suivant : **cesarE1&2-J2.py**

#### 4.4 Implémentation avec utilisation du code Ascii

**Question 14.** Faire une copie du fichier dont le nom est `cesarE1&2-J2.py`.

**Question 15.**

- **Créer** un fichier texte, dont le nom est `alphabet.txt`, contenant toutes les lettres de l'alphabet.
- **Stocker** l'ensemble des lettres de l'alphabet dans une variable de type chaîne de caractère et nommée **alphabet**. Pour cela utiliser la fonction **lireFichier()** déjà écrite précédemment.
- En vous aidant de l'algorithme donné, écrire une fonction qui permet de remplacer chaque lettre du texte par son code Ascii :
  - Nom de la fonction : **carAsciiCode()**
  - Paramètres : **listeLettres** et **cle**
  - Renvoi : variable « **listeCodeAscii** » de type liste contenant l'alphabet décalé en fonction de la valeur de la clé de codage
  - La liste **listeCodeAscii** est stockée dans une variable de type liste de même nom

**Aide :** vous pouvez utiliser les deux fonctions suivantes :

- `ord()`
- `chr()`

Algorithme
<b>DEBUT DE L'ALGORITHME</b> <b>Créer</b> une liste vide nommée <b>listeCodeAscii</b> <b>POUR</b> chaque caractère <b>car</b> de la liste contenant toutes les lettres <b>listeLettres</b> <b>DEBUT POUR</b> <b>SI</b> le code Ascii de <b>car</b> augmenté de la clé est inférieur ou égal au code ascii du caractère z <b>DEBUT SI</b> <b>Ajouter</b> à la liste <b>listeCodeAscii</b> le nouveau code Ascii <b>FIN SI</b> <b>SINON</b> <b>DEBUT SINON</b> <b>Ajouter</b> à la liste <b>listeCodeAscii</b> le nouveau code Ascii <b>FIN SINON</b> <b>FIN POUR</b> <b>RETOURNER</b> la liste contenant le texte codé <b>FIN DE L'ALGORITHME</b>

**Question 16.** **Tester** la fonction.

**Question 17.** **Ecrire** une fonction qui permet de reconstituer le texte codé dans une variable de type chaîne de caractères :

- Nom de la fonction : **reconsTexte()**
- Paramètres : **listeCodeAscii**
- Renvoi : variable « **texteCode** » de type liste contenant l'alphabet décalé en fonction de la valeur de la clé de codage
- La liste **texteCode** est stockée dans une variable de type chaîne de caractère de même nom

**Question 18.** **Ecrire** le texte codé dans un fichier texte dont le nom est **texteMystere.txt**.

**Question 19.** **Tester** l'ensemble.

**Question 20.** **Créer** un fichier compacté au format **cesarE2-J3.zip** contenant tous les fichiers nécessaires au fonctionnement de votre programme et le **rendre** via l'ENT.

Jalon

Etape 1

Etape 2