Mini-projet : un peu de cryptographie ?

Partie 1 : révision flash

L'objectif de cette partie est de vous assurer que vous avez bien acquis les concepts élémentaires de l'algorithmique et du langage Python.

1- Soient les deux variables suivantes :

a=123
b='123'

Quelle est la différence entre ces deux variables ?

Les variables suivantes sont-elles correctes ?

c=a+1
d=b+1

2- soit la liste suivante **L=[1,2,3,4,5,6,7,8,9,10]**

Quelle est la valeur de **L[5]** ?

Ecrire un programme en Python qui utilise une boucle **for** pour afficher les éléments de la liste L

3- Ecrire un programme en Python qui utilise une boucle for pour afficher les 5 derniers éléments de la liste L

4- Ecrire un programme qui utilise une boucle **while** pour afficher les éléments de la liste L depuis le début et qui s'arrête lorsqu'elle trouve une valeur égale à 7.

5- utiliser l'instruction conditionnelle **if** pour afficher pour chaque valeur de la liste L si elle est paire ou impaire.

Partie 2 : les fichiers

Dans cette partie, nous allons introduire les commandes utiles à l'écriture et la lecture dans des fichiers. Les fichiers sont omniprésents en informatique et vous en utilisez un grand nombre quotidiennement. Ils servent par exemple à enregistrer un texte rédigé à l'aide d'un traitement de texte, des données numériques saisies à l'aide d'un tableur, une image, une vidéo, etc.

Dans la suite nous travaillerons avec des fichiers textes simples (c'est-à-dire qu'il permettent d'enregistrer des textes sans formatage particulier).

En langage python, il est possible d'écrire dans un fichier texte grâce à la commande suivante :

f1=open("message1.txt","w", encoding='utf8')

f1 est le nom donné au flux de données (vous êtes libres de choisir n'importe quel nom).

message1.txt est le nom du fichier dans lequel nous allons écrire. S'il en existe un avec le même nom dans le dossier, il sera écrasé !

w indique que nous ouvrons ce fichier en écriture (write)

encoding='utf8' permet d'enregistrer les caractères accentués (code ASCII étendu)

Voilà ! vous venez d'ouvrir le fichier (ou de le créer s'il n'existe pas) ! Il faut maintenant écrire dedans !

Pour écrire dans un fichier ouvert, vous pouvez utiliser par exemple la commande :

f1.write('Mon texte')

Si vous souhaitez écrire plusieurs lignes incluant un retour à la ligne, vous pouvez par exemple procéder ainsi :

f1.write('PREMIERE LIGNE\n')

f1.write('SECONDE LIGNE')

Notez le **\n** (retour à la ligne) à la fin de la première ligne.

Une fois l'écriture terminée il convient de fermer le fichier sans quoi votre fichier risque d'être corrompu et son contenu illisible. La fermeture d'un fichier se fait à l'aide de la commande :

f1.close()

Pour lire le contenu d'un fichier texte, il est possible de procéder ainsi pour l'ouvrir en lecture :

f1=open("message1.txt","r", encoding='utf8')

Quelle est la différence avec la commande d'ouverture en écriture ? Vous avez certainement trouvé !

Il est alors possible de lire le contenu du fichier en entier à l'aide de la commande :

s=f1.read()

Dans le cas d'un fichier multilignes, on peut utiliser la commande :

s=f1.readlines()

1- Ecrire un programme qui crée un fichier nommé source.txt et qui contient le message suivant :

« SUPER ! Je viens de créer mon premier fichier ! »

2- Ecrire un programme qui recopie le contenu du fichier source.txt dans le fichier destination.txt

3-Expérimentez l'écriture et la lecture de plusieurs lignes.

Partie 3 : encore une dernière chose : le code ASCII

Comme vous le savez, un ordinateur ne manipule que des données binaires, c'est-à-dire des « 0 » et des « 1 ». Lors de la transmission de caractères, ces derniers sont transformés en données numériques grâce au code ASCII. Le code ASCII a, par la suite, été étendu pour contenir également les caractères accentués et spéciaux des différentes langues. Etudier le tableau du code ASCII fourni.

En Python, il est possible d'avoir le code ASCII d'un caractère de manière simple. Par exemple :

```
x='A'
y=ord(x)
print(y)
```

Qu'affiche ce programme ?

Il est également possible d'obtenir le caractère correspondant à un code ASCII donné. Par exemple :

```
x=55
y=chr(x)
print(y)
```

Qu'affiche ce programme ?

1- Expérimentez les conversions entre caractères et équivalent ASCII.

Partie 4 : Cryptons, cryptons !

Vous possédez à présent toutes les compétences pour débuter dans l'espionnage et le cryptage des données !

1- La façon la plus simple (mais qui n'est pas très efficace) de coder un texte est de remplacer chaque caractère par un autre caractère. Par exemple, si le procédé de cryptage consiste à augmenter le code ASCII de chaque caractère d'une unité, le texte ABCDEF deviendra BCDEFG. En effet :

ABCDEF correspond aux codes ASCII **65 66 67 68 69 70**

BCDEFG correspond aux codes ASCII **66 67 68 69 70 71**

Ecrire un programme permettant de lire le contenu du fichier source.txt, de le coder comme indiqué ci-dessus et de sauvegarder le texte résultant dans le fichier **message_code.txt**

2- Le procédé décrit ci-dessus peut être complexifié en choisissant un décalage de plus d'une unité.

Sachant que le message fourni dans le fichier **message_secret.txt** a été décalé d'une valeur comprise entre -4 et +4, écrire un programme qui permet de découvrir le message secret.

3- Compléter votre programme pour sauvegarder le message en clair dans le fichier **message_decode.txt**

Si vous arrivez jusqu'au bout du sujet, vous méritez votre carte d'agent secret !