LMI 1 - Semestre 2

Université d'Artois
Faculté des Sciences Jean Perrin

ALGO2 – Algorithmique et Programmation en Python 2

Fiche de TP numéro 1

La librairie

Exercice 1 : On s'intéresse à la gestion d'un catalogue de livres. Chaque livre est caractérisé par son auteur, le titre de l'œuvre, son année de sortie, et son prix en euros.

```
livre1 = {"Auteur": "Sartre", "Annee":19964, "Titre": "Les mots","Prix":13.5, "Quantité":20}
livre2 = {"Auteur": "Simon Veil", "Annee":2007, "Titre":Une vie":, "Prix":15, "Quantité":35}
livre3 = {"Auteur": "Rousseau", "Annee":1770, "Titre":Les confessions":, "Prix":13.5, "Quantité":45}
livre4 = {"Auteur": "Zola", "Annee":1861, "Titre":Perrette":, "Prix":11.5, "Quantité":51}
livre5 = {"Auteur": "De Musset", "Annee":1849, "Titre":Louison":, "Prix":11, "Quantité":34}
```

Un catalogue associe à chaque genre littéraire la liste des livres disponibles.

```
catalogue = {"Autobiographie":[livre2, livre3], "Roman":[livre1],"Pièces de théâtre":[livre4, livre5]}
```

Q 1 . Spécifiez puis écrivez la fonction `ajoute_livre(auteur,annee, titre, prix,quantite , genre, catalogue)` qui permet d'ajouter un livre au catalogue.

Q 2 . Spécifiez puis écrivez la fonction la fonction `estPresent(catalogue,titre)` qui teste l'existence d'un livre dans le catalogue.

```
>>> estPresent(catalogue, "Les mots")
True
```

Q 3 . Spécifiez puis écrivez la fonction `affiche_livre(livre)` qui permet d'afficher un livre.

```
>>> affiche_livre(livre1)
"Auteur": "Jean-Paul Sartre"
"Annee":19964,
"Titre": "Les mots"
"Prix":13.5,
"Quantité":1020
```

Q 4 . Spécifiez puis écrivez la fonction `changer_prix(livre,prix)` qui permet de modifier le prix d'un livre.

```
>>> changer_prix(livre1,10)
>>> affiche_livre(livre1)
"Auteur": "Jean-Paul Sartre"
"Annee":19964,
"Titre": "Les mots"
"Prix":10,
"Quantité":1020
```

Q 5 . Spécifiez puis écrivez la fonction `ajoute_quantite(catalogue,titre,qte)` qui permet d'augmenter le nombre d'exemplaires d'un livre par `qte`.

Q 6 . Spécifiez puis écrivez la fonction `livres_auteur(catalogue,nom_auteur)` qui renvoie la liste des titres des livres du même auteur présents dans le catalogue.

Q 7 . Spécifiez puis écrivez la fonction `livres_annee(catalogue,date)` qui renvoie la liste des livres parue à une date donnée.

Q 8 . Spécifiez puis écrivez la fonction `les_plus_chers(catalogue)` qui renvoie la liste des livres les plus chers du catalogue.

Q 9 . Spécifiez puis écrivez la fonction `genres_littraires(catalogue,nom_auteur)` qui renvoie la liste des genres littéraires d'un auteur.

Q 10 . Spécifiez puis écrivez la fonction `commande(catalogue, liste_livres)` qui permet de tester si une commande de livres a bien pu avoir lieu ou non. Si oui, le catalogue doit être modifié en conséquence. Un livre avec un nombre d'exemplaires égale à 0 doit être supprimé du catalogue.

Message secret

On désire crypter un texte par une méthode très simple : on établit une table de permutation associant à chaque lettre majuscule une autre majuscule.

Vous devez écrire et générer la documentation html de votre module.

Exercice 2 : Spécifier puis écrire une fonction `permuteListe(uneListe)` qui échange successivement chaque élément de la liste (l'élément au rang 0, puis au rang 1, puis ...) avec un autre élément de la liste choisi au hasard. La liste doit pouvoir contenir n'importe quel type d'élément. La liste donnée en paramètre est modifiée par cette fonction

Exemple :

```
> l = [ 'A', 'B', 'C', 'D' ]
> permuteListe(l)
['B', 'C', 'A', 'D']
> l
['B', 'C', 'A', 'D']
```

Exercice 3 : À l'aide de la fonction précédente, spécifier puis écrire une fonction `dictPerm()` qui fabrique et retourne un dictionnaire de permutation (tiré au hasard) pour les lettres majuscules.

Exemple d'utilisation :

```
> perm = dictPerm()
> perm
{'Z': 'A', 'X': 'N', 'Y': 'E', 'V': 'J', 'W': 'H', 'T': 'L', 'U': 'C',
 'R': 'O', 'S': 'Q', 'P': 'V', 'Q': 'Y', 'N': 'M', 'O': 'U', 'L': 'F',
 'M': 'K', 'J': 'I', 'K': 'R', 'H': 'T', 'I': 'B', 'F': 'Z', 'G': 'W',
 'D': 'P', 'E': 'G', 'B': 'X', 'C': 'D', 'A': 'S'}
```

Pour cette question, vous utiliserez la variable `ascii_uppercase` du module `string` qui contient la chaîne de caractères des lettres de l'alphabet en majuscules :

```
> import string
> string.ascii_uppercase
'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

Exercice 4 : Spécifier puis écrire la fonction `crypte(txt, perm)` qui crypte le texte `txt` à l'aide du dictionnaire de permutation `perm`.

Exercice 5 : Pour décoder le texte crypté, il faut appliquer les mêmes permutations à l'envers. Écrire une fonction `invertDict(perm)` qui renverse un dictionnaire. La fonction retourne un autre dictionnaire, comme dans cet exemple :

```
> D = { 'A' : 'C', 'B' : 'A', 'C' : 'B' }
> invertDict(D)
{'B': 'C', 'C': 'A', 'A': 'B'}
```

Exercice 6 : Spécifier puis écrire la fonction `decrypte(txt, perm)` qui décrypte le texte `txt` qui a été crypté à l'aide du dictionnaire de permutation `perm`.

Exercice 7 : Vous devez maintenant écrire un programme principal qui permet de crypter et décrypter des textes. Voilà un exemple d'interaction :

```
moi@maMachine:~/algo2/semaine2$ python3 codage.py
1 : Liste des codes disponibles.
2 : Créer un nouveau code.
3 : Crypter un texte.
4 : Décrypter un texte.
0 : Quitter l'application.
Votre choix : 2
Donnez le nom de votre nouveau code : prems
Le nouveau code est : {'U': 'N', 'D': 'V', 'E': 'G', 'F': 'J', 'M': 'U'...}
1 : Liste des codes disponibles.
2 : Créer un nouveau code.
3 : Crypter un texte.
4 : Décrypter un texte.
0 : Quitter l'application.
Votre choix : 2
Donnez le nom de votre nouveau code : deuz
Le nouveau code est : {'U': 'H', 'D': 'O', 'E': 'E', 'F': 'J', 'M': 'W'...}
1 : Liste des codes disponibles.
2 : Créer un nouveau code.
3 : Crypter un texte.
4 : Décrypter un texte.
0 : Quitter l'application.
Votre choix : 3
Liste des codes disponibles :
prems : {'U': 'N', 'D': 'V', 'E': 'G', 'F': 'J', 'M': 'U', 'J': 'H',...}
deuz : {'U': 'H', 'D': 'O', 'E': 'E', 'F': 'J', 'M': 'W', 'J': ' ',...}
--
Nom du code choisi : prems
Texte à crypter : bonjour a tous
Le texte crypté est :
QLSHLNMDPDFLNI
```

```
1 : Liste des codes disponibles.
2 : Créer un nouveau code.
3 : Crypter un texte.
4 : Décrypter un texte.
0 : Quitter l'application.
Votre choix : 4
Liste des codes disponibles :
... <ici les codes disponibles>
--
Nom du code à choisir : prems
Texte à décrypter : QLSHLNMDPDFLNI
Le texte décrypté est :
BONJOUR A TOUS
1 : Liste des codes disponibles.
2 : Créer un nouveau code.
3 : Crypter un texte.
4 : Décrypter un texte.
0 : Quitter l'application.
Votre choix : 7
Votre choix : 0
Bye
```

Vous devez écrire une fonction pour chacune des actions du menu :

Q 1 . Spécifier et écrire la fonction `affiche_menu` qui affiche toutes les options du menu principal et retourne le choix de l'utilisateur.

Q 2 . Spécifier puis écrire la fonction `afficher_les_codes` qui, à partir d'un dictionnaire qui, à un nom, associe un code, affiche tous les codes disponibles.

Q 3 . Spécifier puis écrire la fonction `ajouter_un_code` qui prend en paramètre le dictionnaire de tous les codes et en créé un nouveau.

Q 4 . Spécifier puis écrire les fonctions `crypter_un_texte` et `decrypter_un_texte`.

Q 5 . Écrire le script principal. Comment feriez-vous pour prendre en compte les espaces dans un texte ? les minuscules et les majuscules ?