TP 2. Séquences et dictionnaires.

intro...

Le but de ce TP est blablabla.

Créez un dossier `TP02` puis, à l'intérieur, un fichier `TP02.py`.

REMARQUE

Le signe ♯ est utilisé devant un commentaire. Pour chaque exercice, vous devrez en entête inscrire le titre de l'exercice.
Les tests effectués seront conservés sous forme de commentaires dans le script de votre programme. Le signe de commentaire ♯ en début de ligne peut être mis en sélectionnant les lignes à commenter et en choisissant « comment out region » dans l'onglet format (pour décommenter, vous pouvez utiliser « uncomment region »).
On rappelle aussi qu'il est préférable de faire les tests dans le script (au moyen de `print`) plutôt que dans le shell. Il suffit ensuite de mettre en commentaire les `print`.

1 Travail sur les listes

2 Déterminer la fréquence d'apparition des lettres dans un texte

Récupérez le fichier `texte.py` sur le site `changer le nom du site`, et enregistrez-le dans votre dossier `TP02`. Après avoir ouvert et parcouru `texte.py`, copiez-collez son contenu au début de votre fichier `TP02.py`.

On rappelle que c'est le type `string` (chaîne de caractères) qui permet, en python, de manipuler les caractères et les textes : par exemple `'a'` et `'bonjour, ça va ?'` sont de type `string`.

La fonction `compterLettre(texte)` est donnée,

```
def compterLettre(texte:str):
     dictionnaire={}
     for lettre in texte:
          if lettre in 'azertyuiopqsdfghjklmwxcvbn':
               if lettre in dictionnaire.keys():
                    dictionnaire[lettre]+=1
               else:
                    dictionnaire[lettre]=1
     return dictionnaire
```

Question 1. Commenter chaque ligne de la fonction `compterLettre` et expliquer ce qu'elle fait.

Pour connaître la fréquence d'apparition des lettres dans un texte, nous allons compter le nombre de lettres dans le texte à partir du dictionnaire obtenu avec la fonction `compterLettre(texte)` (sans ponctuation ni caractère particulier),

Question 2. Écrire la fonction `sommeLettre(dico:dict)` qui prend comme argument un dictionnaire dont les clés sont les lettres de l'alphabet et les valeurs le nombre d'apparition de chaque lettre dans le texte considéré. Cette fonction renvoie un entier, somme de toutes les lettres.

Question 3. Écrire la fonction `frequenceLettre(dico:dict)` qui prend pour argument le même dictionnaire que dans la fonction `sommeLettre` et qui renvoie un dictionnaire dont les clés sont les lettres de l'alphabet et les valeurs les pourcentages d'apparition des lettres dans le texte considéré. Vous pourrez avantageusement utiliser la fonction `sommeLettre` dans votre fonction.

Vous devez trouver pour la lettre 'i', 8.009153318077804%

3 Gérer les stocks de composants pour réaliser des drones

L'entreprise `trucMuch`...

Les composants utiles pour réaliser un drone sont :

- le châssis ;
- les 4 moteurs ;
- les 4 hélices ;
- la batterie ;
- le contrôleur de vol ;
- ESC 4 en 1 pour les 4 moteurs (Electronic Speed Controler) ;
- la plaque de distribution de puissance (PDB Power Distribution Board).

Des composants en option sont aussi disponibles (caméra, radiocommande, chargeur, buzzer, leds...) et ne seront pas traités ici.

Dans le fichier `drone.py` donné, vous trouverez 4 dictionnaires dont les clés sont les noms des composants et les valeurs représentent le nombre de composants nécessaire :

```
drone={'chassis':1,'moteur':4,'helice':4,'controleurVol':1,'ESC4en1':1,'batterie':1,
'plaqueDeDistribution':1}
stock={'chassis':0,'moteur':25,'helice':36,'controleurVol':12,'ESC4en1':8,'batterie':20,
'plaqueDeDistribution':7}
limiteMin={'chassis':2,'moteur':8,'helice':8,'controleurVol':2,'ESC4en1':2,'batterie':2,
'plaqueDeDistribution':2}
limiteMax={'chassis':15,'moteur':60,'helice':60,'controleurVol':15,'ESC4en1':15,'batterie':30,
'plaqueDeDistribution':15}
```

- `drone`, correspond aux composants nécessaires à la réalisation d'un drone. Les clés étant les composants et les valeurs le nombre de composant pour un drone ;
- `stock`, correspond au stock à l'instant considéré ;
- `limitMin`, correspond aux valeurs limites basses du stock pour déclencher une commande ;
- `limitMax`, correspond aux valeurs limites hautes pour reconstituer le stock et définir le nombre de composants de la commande.

Récupérez le fichier `drone.py` sur le site `changer le nom du site`, et enregistrez-le dans votre dossier `TP02`. Après avoir ouvert et parcouru `drone.py`, copiez-collez son contenu au début de cette activité dans votre fichier `TP02.py`.

OBJECTIF

Écriture d'un programme qui permette de générer les commandes de composants utiles pour assurer la réalisation des drones attendus par les clients sans avoir de rupture de stock.
On utilisera des objets de type `dict`.

Réaliser un drone

Question 1. Écrire la fonction `realiser1Drone(drone:dict, stock:dict)` qui prend pour argument les dictionnaires `drone` et `stock` et qui renvoie un booléen, `True` si le stock est suffisant pour réaliser un drône sinon `False`.

Gérer le stock de composants

Question 2. Écrire la fonction `destocker(drone:dict, stock:dict)` qui prend pour argument les dictionnaires `drone` et `stock` et qui retire du stock le nombre de composants utiles pour réaliser un drône. Cette fonction ne renvoie rien. Le dictionnaire `stock` est modifié par effet de bord.

Lorsque le stock devient insuffisant, une commande est passée et le stock est ré-évalué.

Question 3. Écrire la fonction `stocker(commande:dict, stock:dict)` qui prend pour argument les dictionnaires `commande` et `stock` et qui ajoute au stock le nombre de composants commandés. Cette fonction ne renvoie rien. Le dictionnaire `stock` est modifié par effet de bord.

Passer une commande

Une commande est passée quand le stock d'un composant est à la limite basse (valeur incluse). Les composants qui n'ont pas atteint la limite basse ne sont pas commandés.

Question 4. Écrire la fonction `commanderComposant(stock:dict,limiteMin:dict,limiteMax:dict)` qui prend pour argument les dictionnaires `stock`, `limiteMin` et `limiteMax` et qui renvoie un dictionnaire `commande` permettant de reconstituer le stock.

Gestion automatique

Dans la semaine, l'entreprise trucmuch reçoit les commandes de drones de 4 clients sous la forme d'une liste, `listeCommande`=`[3,1,5,2]`.

Question 5. Écrire la fonction `satisfaireClient(listeCommande:list,drone:dict,stock:dict, limiteMin:dict,limiteMax:dict)` qui prend pour argument une liste de commande de drones, les dictionnaires `drone`, `stock`, `limiteMin` et `limiteMax` et qui affiche l'état du stock après chaque réalisation d'un drone ainsi que les commandes successives.