Numérique et science informatique
Classe de terminale 2024-25
Contrôle 2
Jeudi 14 novembre 2024

Consignes

- Rédiger sur feuilles doubles avec :
  * un cadre pour la notation et les commentaires
  * et une marge d'au moins 4 cm.
- Seules les réponses écrites sur votre copie seront prises en compte. Toutes les réponses doivent être justifiées.
- Le sujet à rendre avec la copie.
- La calculatrice n'est pas autorisée.
- Le barème est indicatif.

Exercice 1 : Dictionnaires (8 points)

La cryptographie est un ensemble de techniques permettant de chiffrer un message.

Une technique de cryptographie consiste à mélanger les lettres d'un alphabet et à réécrire le message avec ces permutations. En Python, on peut créer un dictionnaire dans lequel les clés sont les lettres de l'alphabet et les valeurs sont celles de l'alphabet mélangé.

Par exemple, si l'alphabet contient les 4 lettres A, B, C et D, et si le dictionnaire de l'alphabet mélangé est `alpha ={"A": "B", "B": "D", "C": "A", "D": "C"}`, la chaîne de caractères `"BAC"` sera chiffrée `"DBA"`.

Un tel dictionnaire sera appelé **dictionnaire de chiffrement**.

1. On souhaite chiffrer un message écrit avec l'alphabet `A, B, C, D, E, F, G` à l'aide du dictionnaire `alpha ={"A":"B", "B":"D", "C":"A", "D":"C", "E":"F", "F":"G", "G":"E"}`

a. Quelle est la valeur associée à la clé `"D"` ? En Python, comment l'obtenir ?

b. Chiffrer la chaîne de caractères `"BAGAGE"` avec le dictionnaire `alpha`.

2. On considère qu'un mot est une chaîne de caractères (un objet de type `str`) écrite uniquement avec les 26 lettres de l'alphabet en majuscule. Par exemple, `"ARBRE"` est un mot et `"L'ARBRE !"` n'est pas un mot à cause des caractères : `"'"`, `" "`(espace) et `"!"`.
Écrire une fonction `chiffrer(mot, alpha)` qui prend en paramètres `mot` un mot et `alpha` un dictionnaire de chiffrement, et qui renvoie une chaîne de caractères chiffrée avec le dictionnaire de chiffrement `alpha`.

3. On souhaite déchiffrer un mot chiffré avec cette méthode.

a. Si un mot est chiffré avec le dictionnaire de chiffrement `alpha ={"A":"B", "B":"D", "C":"A", "D":"C", "E":"F", "F":"G", "G":"E"}`, donner un dictionnaire permettant de le déchiffrer.

b. Écrire une fonction en Python appelée `dico_dechiffrement(dico)` qui prend en paramètre `dico` un dictionnaire de chiffrement et qui renvoie un dictionnaire

permettant le déchiffrement. On pourra s'inspirer du code incomplet ci-dessous ou proposer une autre solution :

```
def dico_dechiffrement(dico):
    nouveau = {}
    for lettre in dico :
        code = dico[...]
        nouveau[...] = ...
    return nouveau
```

c. Écrire une fonction `déchiffre(mot, dico)` qui reçoit un mot chiffré et un dictionnaire de chiffrement et renvoie le mot décodé. On utilisera les fonctions écrites dans les questions précédentes.

4. On souhaite à présent créer un dictionnaire de chiffrement. Ecrire une fonction `dico_chiffrement(alphabet)` qui prend en paramètre `alphabet` un tableau de lettres et qui renvoie un dictionnaire de chiffrement dont les clés sont les lettres du tableau `alphabet` et les valeurs sont les lettres du tableau `alphabet` mélangées.

On pourra utiliser la fonction `shuffle` du module `random` qui mélange en place un tableau. Par exemple, on a :

```
>>> tab = ["A", "B", "C", "D"]
>>> shuffle(tab)
>>> tab
["B", "A", "D", "C"]
```

Exercice 2 : Programmation orientée objet (12 points)

Une entreprise, présente sur différents sites en France, attribue à chacun de ses employés un numéro de badge unique.

Dans le tableau ci-dessous, on donne le numéro de badge, le nom, le prénom et les années de naissance et d'entrée dans l'entreprise de quelques salariés.

numéro de badge	nom	prénom	année de naissance	année d'entrée
112	LESIEUR	Isabelle	1982	2005
2122	VASSEUR	Adrien	1962	1980
135	HADJI	Hakim	1992	2015

Pour chaque personne, on souhaite stocker les informations dans un objet de la classe `Personne` définie ci-dessous :

```
class Personne():

    def __init__(self, num, n , p , a_nais, a_ent):
        self.num_badge = num
        self.nom = n
        self.prenom = p
        self.annee_naissance = a_nais
        self.annee_entree = a_ent
```

1. Indiquer le nom et le type de tous les attributs d'un objet de la classe `Personne`.

2. Écrire à l'aide du tableau précèdent, les trois instructions permettant de créer les objets `personneA`, `personneB` et `personneC` contenant les informations pour les trois personnes du tableau : LESIEUR Isabelle, VASSEUR Adrien et HADJI Hakim.

3. Donner l'instruction permettant d'obtenir dans la variable `numA` le numéro de badge de l'objet `personneA` instancié à la question précédente.

On souhaite ajouter une méthode `annee_anciennete` à la classe `Personne` qui donne le nombre d'années d'ancienneté d'une personne au sein de l'entreprise. Par exemple : Madame LESIEUR Isabelle a une ancienneté dans l'entreprise de 19 ans en considérant que nous sommes en 2024.

4. Recopier et compléter le code suivant de la méthode `annee_anciennete` :

```
    def annee_anciennete(self):
        return ...
```

On considère la classe `Personnel` qui modélise la liste du personnel d'une entreprise et dont le début de l'implémentation est la suivante :

```
class Personnel:
    def __init__(self):
        self.liste = []
```

5. a. Écrire la méthode `ajouter` permettant d'ajouter un objet de type `Personne` à la liste du personnel de l'entreprise de la classe `Personnel`.

b. Écrire deux instructions qui :

* crée un objet de la classe `Personnel` et l'affecte à la variable `perso_entreprise` ;

* utilise la méthode `ajouter` pour ajouter `personneA` à la liste du personnel de l'objet `perso_entreprise`.

6. a. Écrire la méthode `effectif` de la classe `Personnel`. Cette méthode devra renvoyer le nombre de personnes présentes dans l'entreprise.

b. Écrire l'instruction qui utilise la méthode `effectif` pour affecter à la variable `eff` l'effectif du personnel de l'entreprise représenté dans la variable `perso_entreprise`.

7. Recopier et compléter la méthode `donne_nom` de la classe `Personnel`. Cette méthode prend en paramètre le numéro de badge d'une personne et renvoie le nom de la personne correspondant à ce badge si elle existe, ou `None` sinon.

```
    def donne_nom(..., num):
        for elt in self.liste :
            if ... == num:
                return ...
        return ...
```

8. Lors de la cérémonie des vœux, l'entreprise souhaite mettre à l'honneur les personnes ayant exactement 10 ans d'ancienneté dans l'entreprise. Écrire une méthode `nb_personne_honneur` de la classe `Personnel` qui prend en paramètre l'année de la cérémonie et qui retourne le nombre de personne(s) à mettre à l'honneur.

9. Écrire une méthode `plus_anciens` de la classe `Personnel` qui retourne la liste des numéros de badge des personnes ayant la plus grande ancienneté dans l'entreprise.