Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université de Sfax
Institut Préparatoire aux Etudes
d'Ingénieurs de Sfax

IPEIS

وزارة التعليم العالي والبحث العلمي
جامعة صفاقس
المعهد التحضيري للدراسات الهندسية
بصفاقس

Année universitaire : 2021-2022

Examen semestre 2

Epreuve d'informatique

Classe : MP2, PC2 et PT2 **Nombre de pages** : 4

Date : Mai 2022 **Durée** : 2h

Note : L'annexe du sujet contient des rappels de Python.

Exercice 1 :

Le but de cet exercice est de gérer des SMS. Un fichier de SMS brut est un texte de la forme suivante :

54342310	31/05/2016	10h23	Salut, t'es ou ?
98304059	30/05/2016	11h10	Sonia t'a laissé un message. Rappelle le : 99432340
54394501	01/06/2016	09h03	T'as pas la solution de l'exo 2 ?
54394503	01/06/2016	09h40	Qu'est-ce que tu fais, y en a marre d'attendre.
27324524	29/05/2016	06h30	Rdv dans 5 min au 89 bvd Royal.
53434325	28/04/2016	03h20	Le nouveau IPHONE à 400 TND chez phonehouse.
23523523	03/06/2016	16h23	ADIDAS: PLUS QUE 20 MODELES DISPO CHEZ SPORT GO.
85750000	29/08/2016	13h31	Envoyer OK au 85750000 pour recevoir un cadeau.

Chaque ligne contient exactement un SMS. Les numéros de téléphone ont tous 8 chiffres, les dates sont toujours formées de 10 caractères et l'heure de 5. Les numéros de téléphone, les dates, l'heure et le texte du message sont séparés par une tabulation \t.

Un dictionnaire de SMS est composé de clefs qui sont les numéros de téléphone d'origine et de valeurs qui sont des listes. Chaque liste contient 3 éléments [date,heure,contenu].

On considère qu'un SMS est publicitaire s'il contient la chaine 'TND' ou si plus que la moitié des caractères sont des majuscules. On considère qu'un message est un SPAM si le numéro de téléphone est 85750000.

N.B : dans les réponses aux questions, si un traitement a été réalisé dans une fonction des questions précédente, il faut l'appeler et non le reproduire.

1. Écrire une fonction **lire_sms** ayant comme argument le nom d'un fichier de SMS brut, renvoie un dictionnaire de SMS.
2. Écrire une fonction **no_spam** qui, à partir d'un dictionnaire de SMS, renvoie un dictionnaire sans les SMS SPAM.

3. Écrire une fonction qui élimine d'un dictionnaire de SMS tous les SMS publicitaires.

4. Écrire une fonction qui extrait d'un dictionnaire de SMS une liste de numéros de téléphone qui ont envoyé soit des SMS publicitaires soit des SMS SPAM.

5. Écrire une fonction qui prend en argument un dictionnaire de SMS et une chaine de caractères. Cette fonction doit retourner une liste de tous les numéros qui ont envoyé un message contenant la chaine.

6. Écrire une fonction qui étant donné un dictionnaire de SMS et une date renvoie un dictionnaire de SMS sans les SMS qui sont plus vieux que 2 semaines.

Exercice 2 :

1. Écrire une fonction **SList(L,i)** qui prend en argument une liste d'entiers **L** et un entier **i** strictement positif, et qui renvoie la sous-liste **SL** contenant les éléments de **L** qui sont divisibles par **i**.

2. Écrire une fonction **cprod(E,A)** qui prend un ensemble de départ **E** et un ensemble d'arrivé **A** en argument, et qui renvoie l'ensemble de tous les tuples **(x,y)** formés d'un élément **x** de **E** et un élément **y** de **A**.

3. Écrire une fonction **listn(L,i)** qui prend une liste de listes **L** en argument, et renvoie une liste de toutes les sous listes d'éléments de **L** de longueur **i**.

4. Qu'imprime le programme suivant ?

```
def shuffle(a, b, c = []):
    if a == [] :
        return [c + b]
    if b == [] :
        return [c + a]
    return shuffle(a[1:], b, c + [a[0]]) + shuffle(a, b[1:], c + [b[0]])
print(shuffle([1,2],[3,4]))
```

Exercice 3 :

On donne trois tables concernant les gouvernorats, les délégations et les municipalités de Tunisie (une municipalité appartient à une délégation et une délégation à un gouvernorat).

Gouvernorats (num_gouv, nom)

Delegations (num_del, #num_ gouv, nom)

Municipalites (num_muni, #num_del, nom, nbr_habitants, surface)

N.B : tout attribut souligné désigne une clé primaire et tout attribut précédé par # désigne une clé étrangère.

Donner les requêtes SQL permettant de :

1. Donner le nombre total d'habitants de la Tunisie.

2. Donner le nombre de municipalités dans la délégation "Sfax_sud".

Requêtes avec jointures

3. Donner la liste des noms des délégations des gouvernorats "Mannouba" et "Ariana".

4. Donner sans doublons la liste des noms des délégations contenant une municipalité dont le nom commence par "Ben".

Requêtes imbriquées

5. Donner la liste des noms des municipalités dont la population excède 10 fois la population moyenne des différentes municipalités. (AVG calcule la moyenne)

6. Que donne chacune des requêtes suivantes :

a) SELECT DISTINCT Gouvernorats.nom FROM Delegations JOIN Gouvernorats ON Delegations.num_gouv = Gouvernorats.num_gouv WHERE Delegations.nom LIKE 'B%'

b) SELECT nom FROM Gouvernorats

WHERE EXISTS (SELECT * FROM Delegations

WHERE Delegations.num_gouv = Gouvernorats.num_gouv AND Delegations.nom LIKE 'B%')

N.B : La commande EXISTS s'utilise dans une clause conditionnelle pour savoir s'il y a une présence ou non de lignes lors de l'utilisation d'une sous-requête.

Exercice 4 :

1. Définir une classe Telephone() destinée à initialiser les paramètres d'un téléphone et réaliser certaines opérations de base. Le constructeur de cette classe initialisera les valeurs par défaut : numero_serie = '123456789', code_pin = '1234', modèle = 'huawei honor 7', numero = '12345678'.

   Cette classe possède deux méthodes :

   - Une méthode chercher_reseau(self, R) qui affiche le réseau utilisé : si R = 'TT' alors le message est 'Bien venu à TT', si R = 'Or' alors le message est 'bien venu à Or', si R = 'Og' le message est 'bien venu à Og'.
   - Une méthode allumer(self, codePin, R) qui affiche le modèle du téléphone et dans le cas où codePin correspond au code_pin par défaut, affiche le code pin et le réseau utilisé sinon affiche le message 'erreur code pin'.

2. Définir une classe Telephone_photo qui hérite de la classe Telephone. Le constructeur de cette classe effectue, en plus de l'initialisation des paramètres définies par défaut, l'affichage du message 'téléphone+caméra'. Cette nouvelle classe possédera ses propres méthodes : prendre_photo qui affiche le message 'camera ready' et une méthode fonction_test qui affiche le numéro de série et le code pin.

Annexe

Exemples d'utilisation des commandes split et strip

```
>>> ch='bonjour tout le monde \n'
>>> ch.strip()
'bonjour tout le monde'
>>> ch.split()
['bonjour', 'tout', 'le', 'monde']
```

Rappel sur les fonctions de fichiers

- f = open(filename, 'r') : crée un objet fichier de nom logique f du fichier filename.
  - 'r' : ouverture en mode read, le fichier doit exister, s'il n'existe pas une erreur se produit.
  - 'w' : ouverture en mode write, si le fichier existe il l'ouvre écrase le contenu et positionne le curseur au début, sinon le fichier sera créé.
  - 'a' : ouverture en mode ajout "append". Son contenu est conservé. Si le fichier n'existe pas il sera créé.
  - l'option '+' : le fichier est ouvert en lecture et en écriture.
  - l'option 'b' : ouverture d'un fichier binaire.
- f.read() : lit l'ensemble du fichier et le renvoie sous forme de chaîne.
- f.readline() : lit et renvoie une ligne du fichier de f, la fin de ligne (\n) incluse.
- f.readlines() : lit et renvoie une liste de toutes les lignes du fichier de f, où chaque ligne est représentée par une chaîne se terminant par \n.
- f.write(s) : écrit la chaîne s dans le fichier de f.
- f.writelines(lst) : écrit la liste de chaîne lst dans le fichier de f.
- f.close() : ferme le fichier.

Rappel sur les conteneurs

- C=NULL, L=[], E=set(), D={ }, T=(,) : créent respectivement : un conteneur C, une liste L, un ensemble E, un dictionnaire D et un tuple T vide.
- S.union(T) : retourne l'union de S et T avec S et T deux ensembles.
- Un dictionnaire est un objet conteneur associant des clés à des valeurs.
- d= {cle1:valeur1, cle2=valeur2, cleN=valeurN} : crée un dictionnaire d.
- d[cle] = valeur : ajoute ou remplace un élément dans le dictionnaire d
- del d[x] : supprime la clé x et sa valeur.
- d.pop(x) : supprime la clé x et sa valeur.
- d.keys() : renvoie une séquence des clés de d (cette séquence se parcourt comme une liste).
- d.values() : renvoie une séquence des valeurs de d (cette séquence se parcourt comme une liste).
- d.items : renvoie une séquence des clés, valeurs de d.