

Examen SEMESTRE1**Matière : INFORMATIQUE****Classes : 2^{ème} Année BG Durée : 2 heures**

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.
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 chaîne '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 chaîne de caractères. Cette fonction doit retourner une liste de tous les numéros qui ont envoyé un message contenant la chaîne.
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: Données générales sur le monde

On considère la base de données `world` relative aux pays et principales villes du monde. Elle est constituée des relations suivantes :

- `City(Id, Name, CountryCode, District, Population)`. `City` représente la table des villes.
- `Country(Code, Name, Continent, SurfaceArea, Population, GNP, LifeExpectancy, Capital)`. `Country` représente la table des pays.
- `CountryLanguage(CountryCode, Language, isOfficial, Percentage)`. `CountryLanguage` représente la table des langues. L'attribut `Percentage` représente le pourcentage du langage parlé dans ce pays.

Les clés primaires sont soulignées. Des extraits des différentes tables sont fournis en annexe.

Travail demandé :

Dans la suite on suppose que les trois tables de la base "`EXERCICE2.db`" sont créées et remplies. Traduire en langage SQL les requêtes fournissant :

1. le nom et le GNP de tous les pays dont le GNP est supérieur à 100 000 ;
2. le nom et la population de toutes les villes françaises référencées dans la base de données ;
3. le nombre de personnes parlant anglais dans le monde ;
4. la population totale de chaque continent ;
5. le nom de tous les pays qui parlent plus qu'une langue.

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`.

Code	Name	Continent	SurfaceArea	Population	LifeExpectancy	GNP	Capital
DOM	Dominican Republic	North America	48511.00	8495000	73.2	15846.00	587
DZA	Algeria	Africa	2381741.00	31471000	69.7	49982.00	35
ECU	Ecuador	South America	283561.00	12646000	71.1	19770.00	594
EGY	Egypt	Africa	1001449.00	68470000	63.3	82710.00	608
ERI	Eritrea	Africa	117600.00	3850000	55.8	650.00	652
ESH	Western Sahara	Africa	266000.00	293000	49.8	60.00	2453
ESP	Spain	Europe	505992.00	39441700	78.8	553233.00	653
EST	Estonia	Europe	45227.00	1439200	69.5	5328.00	3791
ETH	Ethiopia	Africa	1104300.00	62565000	45.2	6353.00	756
FIN	Finland	Europe	338145.00	5171300	77.4	121914.00	3236
FJI	Fiji Islands	Oceania	18274.00	817000	67.9	1536.00	764
FLK	Falkland Islands	South America	12173.00	2000	NULL	0.00	763
FRA	France	Europe	551500.00	59225700	78.8	1424285.00	2974
FRO	Faroe Islands	Europe	1399.00	43000	78.4	0.00	901
FSM	Micronesia, Federated States of	Oceania	702.00	119000	68.6	212.00	2689
GAB	Gabon	Africa	267668.00	1226000	50.1	5493.00	902
GBR	United Kingdom	Europe	242900.00	59623400	77.7	1378330.00	456
GEO	Georgia	Asia	69700.00	4968000	64.5	6064.00	905
GHA	Ghana	Africa	238533.00	20212000	57.4	7137.00	910
GIB	Gibraltar	Europe	6.00	25000	79.0	258.00	915

(a) Table Country

CountryCode	Language	IsOfficial	Percentage	ID	Name	CountryCode	District	Population
ABW	Dutch	T	5.3	1	Kabul	AFG	Kabul	1780000
ABW	English	F	9.5	2	Qandahar	AFG	Qandahar	237500
ABW	Papiamentu	F	76.7	3	Herat	AFG	Herat	186800
ABW	Spanish	F	7.4	4	Mazar-e-Sharif	AFG	Balkh	127800
AFG	Balochi	F	0.9	5	Amsterdam	NLD	Noord-Holland	731200
AFG	Dari	T	32.1	6	Rotterdam	NLD	Zuid-Holland	593321
AFG	Pashto	T	52.4	7	Haag	NLD	Zuid-Holland	440900
AFG	Turkmenian	F	1.9	8	Utrecht	NLD	Utrecht	234323
AFG	Uzbek	F	8.8	9	Eindhoven	NLD	Noord-Brabant	201843
AGO	Ambo	F	2.4	10	Tilburg	NLD	Noord-Brabant	193238
AGO	Chokwe	F	4.2	11	Groningen	NLD	Groningen	172701
AGO	Kongo	F	13.2	12	Breda	NLD	Noord-Brabant	160398
AGO	Luchazi	F	2.4	13	Apeldoorn	NLD	Gelderland	153491
AGO	Luimbe-nganguela	F	5.4	14	Nijmegen	NLD	Gelderland	152463
AGO	Luvale	F	3.6	15	Enschede	NLD	Overijssel	149544
AGO	Mbundu	F	21.6	16	Haarlem	NLD	Noord-Holland	148772
AGO	Nyaneka-nkhumbi	F	5.4	17	Almere	NLD	Flevoland	142465
AGO	Ovimbundu	F	37.2	18	Arnhem	NLD	Gelderland	138020
AIA	English	T	0.0	19	Zaanstad	NLD	Noord-Holland	135621
ALB	Albanian	T	97.9	20	S-Hertogenbosch	NLD	Noord-Brabant	129170

(b) Table CountryLanguage

(c) Table City