

Exercice 1 : Annuaire

On dispose d'un annuaire stockant pour chaque employé d'une entreprise ses nom et prénom, son numéro de téléphone et le service dans lequel il est employé. Cet annuaire est représenté par un dictionnaire dont les clés sont formées par les tuples (nom, prénom) des employés et les valeurs sont les tuples (téléphone, service) associés.

- 1) Ecrire une fonction qui retourne le numéro de téléphone d'une personne dont on donne le nom et le prénom ou "inconnu dans l'entreprise" dans le cas où cette personne n'apparaît pas dans l'annuaire.
- 2) Ecrire une fonction qui construit la liste de toutes les personnes travaillant dans un service donné.
- 3) Ecrire une fonction qui ajoute une personne dans l'annuaire.

Exercice 2 : Masse atomique

On dispose du tableau périodique des éléments et on souhaite écrire une fonction qui calcule la masse atomique d'un minéral dont on connaît la composition.

Exemple de la composition du carbonate de calcium : $CaCO_3$

TABLEAU PÉRIODIQUE DES ÉLÉMENTS

Numéro atomique → Z ← Masse molaire atomique (g.mol⁻¹)

Symbole atomique → X ← Famille

Nom

Le tableau périodique des éléments est présenté avec les numéros atomiques (Z) et les masses molaires atomiques (M) indiqués. Les symboles atomiques (X) et les familles sont également mentionnés. Les éléments sont classés par groupes (numérotés 1 à 18) et périodes (numérotées 1 à 7).

Exercices

1) Ecrire une telle fonction sachant que le tableau périodique est fourni (dans Arche) sous la forme d'un dictionnaire avec pour chaque élément en clé le symbole atomique et en valeur la masse atomique de cet atome, le minéral est fourni sous forme d'une liste de couples (symbole, qte).

Exemple : `[("Ca", 1), ("C", 1), ("O", 3)]`

2) Modifier la fonction précédente pour utiliser un dictionnaire plus complet (fourni sur Arche), tel que la valeur associée à un symbole est un triplet constitué de sa masse atomique, de son numéro atomique et de son nom complet.

3) Le minéral est en fait fourni sous forme d'une chaîne de caractères, "CaCO3" par exemple. Ecrire une fonction qui analyse cette chaîne de caractères pour la transformer un une liste de composants, par exemple `[("Ca", 1), ("C", 1), ("O", 3)]`, permettant ensuite de pouvoir utiliser la fonction écrite précédemment.

Exemple de minéral : "Na2AlSi3O10H2O"

Indications : on peut utiliser les méthodes `isupper`, `islower` et `isdigit`

Exercice 3 : De la mine à l'usine de traitement

Le (la) responsable du transport des roches extraites d'une mine nous demande de calculer la taille de la benne (en m³) permettant de transporter une tonne de ce matériau

On dispose :

- De la composition de la roche à transporter sous forme d'un tuple de couples (composant, pourcentage massique) (Un couple pouvant être considéré comme un tuple).

Exemple de composition d'une roche de Tuffeau :

- Calcite 50%
- Quartz 10%
- Cristobalite 30%,
- Argile – Mica 10%

- De la masse volumique de chaque minéral en g/cm³ sous forme de dictionnaire (clé=minéral, valeur=masse volumique)

Extrait :

Calcite	2.71	Graphite	2.2
Dolomite	2.9	Magnétite	5.2
Gypse	2.4	Argile	1.7
Quartz	2.65	Cristobalite	2.32

1) Ecrire une fonction qui calcule, à partir de la composition de la roche (tuples des composants) et du dictionnaire des masses volumiques, la masse volumique de la roche extraite de la mine.

2) Ecrire un programme qui affiche le volume (m³) de la benne nécessaire pour transporter une tonne de cette roche.

Rappel : $1 \text{ g/cm}^3 = 1 \text{ kg/dm}^3 = 1 \text{ t/m}^3$

Exercice 4 : Liaisons aériennes

On dispose d'une table associant à chaque ville possédant un aéroport la liste des villes auxquelles elle est reliée. Cette table est stockée sous forme d'un dictionnaire tel que `table[ville_depart] = liste des liaisons [(ville_arrivee, tarif),...]`

1) Écrire une fonction *liaison_directe* qui recherche s'il existe une liaison directe entre deux villes *vdep* et *varr* fournies en paramètre. Cette fonction retourne *True* et le prix de la liaison s'il existe une liaison directe et *False* et 0 sinon.

2) Ecrire une fonction *liaison_indirecte* qui recherche s'il existe une liaison avec une escale **au plus** entre deux villes *vdep* et *varr* fournies en paramètre. Cette fonction retourne un triplet :

True si une liaison avec une escale existe, son prix total et l'escale

False si aucune liaison n'existe, -1, ""

3) Écrire une fonction *liaison* qui recherche s'il existe une liaison directe ou indirecte entre deux villes *vdep* et *varr* fournies en paramètre. Cette fonction retourne :

-1 si une des deux villes n'existe pas dans la table, -1 et ""

1 si une liaison directe existe, son prix et ""

2 si une liaison avec une escale existe, son prix total et l'escale

0 si aucune liaison n'existe, -1, ""

4) Modifier la fonction *liaison_indirecte* pour qu'elle retourne, si elle existe, la liaison indirecte de prix minimal.

Exercice 5 Fréquence de lettres

Ecrire une fonction qui prend en entrée une chaîne de caractères et renvoie un dictionnaire mettant en correspondance chaque caractère apparaissant dans la chaîne avec le nombre de ses occurrences dans la chaîne (fréquence). On comptabilisera de la même façon les minuscules et les majuscules.

Par exemple, pour la chaîne :

"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris a risus. In ornare convallis magna. Nunc tincidunt. Donec pharetra nisi ut felis. Proin iaculis, pede et volutpat imperdiet, felis nulla tincidunt turpis, et sollicitudin risus erat semper lectus."

Exercice 7 : Table des identificateurs

1) Identificateur

On suppose que dans un langage de programmation donné un identificateur est un mot composé de 6 caractères, alphanumériques (lettres et chiffres) ou soulignés "#", commençant obligatoirement par une lettre. Écrire un programme qui lit une suite de caractères un par un et teste si elle forme un identificateur ou non.

Pour ce faire, on pourra écrire des fonctions *lettre(x)* et *chiffre(x)* qui testent respectivement si le caractère x est une lettre ou un chiffre.

2) Table des identificateurs

Dans le langage de programmation étudié, chaque variable doit être déclarée. Dans cet objectif au moment de la compilation, chaque identificateur est rangé dans ce qu'on appelle la *table des identificateurs* qui est en fait un tableau des identificateurs déclarés. On veut construire cette table des identificateurs.

- On suppose qu'on a en plus à notre disposition une table des mots réservés au langage de programmation. Écrire un programme qui lit une suite de mots terminée par le mot FIN, et pour chaque mot lu :

- teste si c'est un identificateur,
- si c'est un identificateur, teste si c'est un mot réservé,
- si c'est un identificateur et pas un mot réservé, et s'il n'appartient pas déjà à la table des Identificateurs le range dans celle-ci,
- si c'est un identificateur et pas un mot réservé, et s'il appartient déjà à la table des identificateurs envoie un message d'erreur de *double déclaration*.

On suppose que la table des mots réservés est donnée et remplie par ordre alphabétique.

Exercice 8 Cryptage - table de substitution

Crypter un texte consiste à le coder de façon à le rendre illisible si on ne connaît pas la méthode utilisée. Une méthode consiste à utiliser une table de substitution. A chaque lettre de l'alphabet on associe une autre lettre de l'alphabet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
T	H	E		Q	U	I	C	K	B	R	O	W	N	F	X	J	M	P	D	V	R	L	A	Z	Y	G

Écrire une fonction qui a en entrée un texte et la table de substitution, et retourne le texte codé de cette façon.

Exercice 9 : Minéraux

On dispose d'un dictionnaire *mineraux* tel que :

`dico_mineraux[nom_mineral] = composition chimique`

Exemple :

`dico_mineraux = {"Gypse" : "Ca-S-O-4.2-H-2-O", "Orthose" : "K-Al-Si-3-O-8", "Calcite" : "Ca-C-O-3", "Diamant" : "C"}`

Les composants et les quantités sont séparés les uns des autres par un "-"

a) Ecrire une fonction :

- Qui reçoit en donnée :
 - un dictionnaire de minéraux tel que présenté ci-dessus
 - un symbole chimique
 - et retourne la liste des minéraux dont la composition chimique intègre ce symbole
- Exemple : pour dico_mineraux et le symbole "Ca" le programme retourne la liste ["Gypse", "Calcite"]*

Indications : On peut utiliser la fonction split de python

```
str.split(sep=None)
Return a list of the words in the string, using sep as the delimiter string.
If sep is given, consecutive delimiters are not grouped together and are
deemed to delimit empty strings (for example, 'toto:alfred'.split(':') returns
['toto', 'alfred']).
```

b) Ecrire une fonction *ajoute_mineral* :

- qui reçoit en entrée :
 - un dictionnaire des minéraux
 - un nom de minéral (par exemple *Sylvanite*)
 - une liste de composants, par exemple `["Ag", "Au", "Te", "4"]`
- et qui ajoute ce minéral dans le dictionnaire (par exemple `"Sylvanite" : "Ag-Au-Te4"`)

Indications :

- *Il faut transformer la liste des composants en une chaîne de caractères, on peut utiliser la méthode join de python*
- *Rappelez-vous les règles de modification d'un dictionnaire dans une fonction*