

## Exercices Corrigés Python (série 7)

### Exercice 1 :

Une molécule est un regroupement d'au moins deux atomes qui sont unis par des liens chimiques et elle est représentée par une formule chimique. Exemple : H<sub>2</sub>O.

Une formule chimique est une succession de symboles d'atomes, suivi chacun par un entier représentant le nombre d'apparitions(**nbr**) de l'atome dans la molécule.

Chaque atome est symbolisé par la première lettre de son nom en majuscule, suivie éventuellement d'une deuxième lettre en minuscule pour distinguer des atomes ayant des initiales identiques. Ainsi, le **Fluor(F)** se distingue de **Fer(Fe)**, du **Fermium(Fm)** et du **Francium(Fr)**.

Le calcul de la masse molaire moléculaire d'une molécule, notée **M(Molécule)**, sera comme suit :

Pour chaque atome de la molécule, calculer le produit (**nbr \* A(atome)**) ou **A(atome)** est un réel représentant la masse atomique de l'atome ;

Calculer la somme des produits obtenus.

### Exemple :

Pour la molécule dichromate de potassium (**K<sub>2</sub>Cr<sub>2</sub>O<sub>7</sub>**) qui est constituée de 2 atomes de potassium(**K**), 2 atomes de chrome(**Cr**) et 7 atomes d'oxygène(**O**), sa masse molaire moléculaire **M(K<sub>2</sub>Cr<sub>2</sub>O<sub>7</sub>)** est égale à **2\*A(K)+2\*A(Cr)+7\*A(O)**.

Puisque A(K)=39,1 g/mol, A(Cr)=52 g/mol et A(O)=16 g/mol, alors M(K<sub>2</sub>Cr<sub>2</sub>O<sub>7</sub>)=2\*39,1+2\*52+7\*16=294,2 g/mol

### Travail demandé

En disposant d'un fichier texte 'Molécules.txt' dont chaque ligne contient le nom d'une molécule suivi de sa formule chimique, séparés par le caractère astérisque '\*'. .

Ecrire une fonction **remplireAtome()** qui permet de remplir le fichier '**Atomes.txt**' par les données relatives à N atomes (N<=50), ou chacun est représenté par son symbole et sa masse atomique,

Ecrire un fonction **masseAtome()** qui permet de stocker dans un fichier 'Resultats.txt' le nom et la masse molaire moléculaire de chaque molécule figurant dans le fichier 'Molecules.txt'.

### Correction :

```
def remplireAtome(N):
    f=open('Atomes.txt','a')
    for i in range(N):
        atome=input('saisir le nom d'atome')
        masse=input('saisir la masse molaire')
        ligne=atome+' * '+masse+' \n'
        f.write(ligne)
    f.close()

def dictionnaire(fichier):
    code=open(fichier)
    dic={}
    for i in code:
```

```

    c=i.strip()
    l=c.split(' ')
    cle=l[0]
    dic[cle]=l[1]
code.close()
return dic

```

```
def massAtome():
```

```

    dic= dictionnaire('Atomes.txt')
    source=open('Molecules.txt')
    dest=open('Resultats.txt','a')
    for ligne in source:
        c=ligne.strip()
        l=c.split('*')
        atome=""
        nb=""
        estnombre=False
        masse=0
        for lettre in l[1]:
            if 'A'<=lettre<='Z' or 'a'<=lettre<='z':
                if estnombre==True:
                    masse+=int(dic[atome])*int(nb)
                    atome=""
                    nb=""
                    estnombre=False
                atome+=lettre
            else:
                estnombre=True
                nb+=lettre
        masse+=int(dic[atome])*int(nb)
    dest.write(l[1]+'!'+str(masse)+'\n')

```

```
massAtome()
```

**Exercice 2 :**

Dans un contexte arithmétique, on définit les nombres premiers factoriels et les nombres premiers primoriels comme indiqué ci-après.

Un nombre PF est dit premier factoriel s'il vérifie les deux propriétés suivantes :

- PF est un nombre premier
- Et PF s'écrit sous la forme d'un factoriel incrémenté ou décrétementé de 1 ( $PF=F! + 1$  ou  $PF=F! - 1$ ), sachant que le factoriel de F noté F ! est égal à  $F*(F-1) * \dots * 1$

**Exemple :**

- 7 est un nombre premier factoriel car 7 est premier et il s'écrit sous la forme  $3! + 1$ .
- 719 est un nombre premier factoriel car 719 est premier et il s'écrit sous la forme  $6! - 1$ .

Un nombre PP est dit premier primoriel s'il vérifie les deux propriétés suivantes :

- PP est un nombre premier
- PP s'écrit sous la forme d'une primorielle incrémentée ou décrétementée de 1 ( $PP=P\#+1$  ou  $PP=P\#-1$ ), sachant que la primorielle de P notée P# est égale au produit des nombres premiers inférieurs ou égaux à P.

**Exemple**

- 211 est un nombre premier primoriel car 211 est premier et il s'écrit sous la forme  $7\# + 1$   
En effet,  $7\#+1=2*3*5*7 + 1 = 210 + 1=211$
- 30029 est un nombre premier primoriel car 30029 est premier et il s'écrit sous la forme  $13\# - 1$   
En effet,  $13\# - 1 = 2*3*5*7*11*13-1=30030 - 1 = 30029$

**Travail demandé :**

1. Ecrire une fonction **premier\_fact(n)** qui permet de vérifier est ce que **n** est un nombre premier factoriel ou non
2. Ecrire une fonction **premier\_primoriel(n)** qui permet de vérifier est ce que **n** est un nombre premier primoriel ou non

**Correction :**

```
def premier(n):
    etat=True
    for i in range(2,(n//2)+1):
        if n%i==0:
            etat=False
            break
    return etat

def premier_fact(n):
    etat=False
    if premier(n):
        f=1
        ordre=2
        while f<n:
```

```
f=f*ordre
ordre+=1
if f+1==n or f-1==n:
    etat=True
return etat

def premier_primorial(n):
    etat=False
    if premier(n)==True:
        p=3
        s=0
        while s<n:
            s=1
            for i in range(2,p+1):
                if premier(i)==True:
                    s=s*i
            p+=1
            if s+1==n or s-1==n:
                etat=True
                break
    return etat

n=719
print(premier_fact(n))
```