

DS 1 - Corrigé

Exercice 1

La variable dico est globale.

1. Les clés sont des chaînes de caractères et la valeurs des listes.

2. `dico[' Pierre..Cox']=[190,76]`

3.

```
def valeur(nom):
    if nom in dico:
        return dico[nom]
    else:
        return []
```

4.

```
def moyennes():
    a,b,n=0,0,0
    for nom in dico:
        t,p=dico[nom]
        a+=t
        b+=p
        n+=1
    return a/n,b/n
```

Exercice 2

On suppose que la liste L est globale (sinon il faut la mettre en paramètre).

1.

```
def creation():
    dico={}
    for (nom,valeur) in L:
        if nom not in dico:
            dico[nom]=[]
        dico[nom].append(valeur)
    return dico
```

2.

```
def clemax(dico):
    n=0
    for nom in dico:
        m=len(dico[nom])
        if m>n:
            n=m
            L=[nom]
        elif m==n:
            L.append(nom)
    return L
```

Exercice 3 (hachage)

1.

```
dico={}
for i in range(26):
    dico[alpha[i]]=1-(i%2)
    dico[Alpha[i]]=1-(i%2)
```

2. (a) $h('Allier') = 8i_0 + 4i_1 + 2i_2 + i_3 = 9$ car les lettres 'A' et 'i' ont un rang impair, et 'l' un rang pair.

De même $h('Ariege') = 8 + 2 + 1 = 11$ car les lettres 'A', 'i' et 'e' ont un rang impair, et 'r' un rang pair.

- (b) Si toutes les lettres ont un rang impair, la valeur obtenue est 15. On obtient tous les entiers de 0 à 15.

3.

```
def hachage(nom):
    s=0
    for i in range(4):
        l=nom[i]
        s+=dico[l]*2**(3-i)
    return s
```

4. (a) On peut avoir des collisions c'est-à-dire deux départements ayant la valeur de hachage. C'est le cas pour les départements 'Paris' et 'Pyrenees'.

```
(b) L=16*['']
    for nom in Dept:
        n=h(nom)
        if L[n]=='':
            L[n]=nom
        else:
            print(' Collision :',L[n]+nom+' avec',L[n])
```

Exercice 4 (Diagramme de Voronoï)

```
1. n=len(Germes)
   for i in range(n):
       plt.plot(Germes[i][0], Germes[i][1], 'x', color=Couleurs[i])
   plt.show()

2. def distance(A,B):
    return ((A[1]-B[1])**2+(A[0]-B[0])**2)**0.5

3. def plusproche(x,y):
    imin=0
    dmin=distance([x,y], Germes[0])
    for i in range(1,len(Germes)):
        d=distance([x,y], Germes[i])
        if d<dmin:
            imin=i
    return Couleurs[imin]

4. for x in range(0,101):
    for y in range(0,101):
        c=plusproche(x,y)
        plt.plot(x,y, 'o', color=c)
    plt.show()
```

Ex1 : 4 (4 × 1). Ex2 : 4 (2 + 2). Ex3 : 7 (5 × 1 + 2). Ex 4 : 5 (1 + 1 + 2 + 1).