INFORMATIQUE : DS2

CORRECTION

EXERCICE 1 - AUTOUR DU MINIMUM

```
def mini(t):
    ''' Calcule le minimum d'un tableau d'entiers ou de flottants'''
    if len(t) == 0:
        return None
    p = t[0]
    for i in range(len(t)):
        if t[i] <= p :
            p = t[i]
    return p
```

1. Faire un tableau, c'est bien.
2. On fait une comparaison par passage en boucle, donc la compléxité est $O(n)$.
3. ```
   def pos_mini(t):
       if len(t) == 0:
           return None
       pos = 0
       for i in range(len(t)):
           if t[i] <= t[pos] :
               pos = i
       return pos
   ```
4. C'est le dernier ici, en faisant `t[i] < t[pos]` on renverrait le premier.
5. ```
   def mini2D(t):
       ''' Calcule le minimum d'une matrice d'entiers ou de flottants'''
       p = t[0][0]
       for i in range(len(t)):
           for j in range(len(t[0])):
               if t[i][j] <= p :
                   p = t[i][j]
       return p
   ```
6. On fait une comparaison par passage dans la double boucle, donc $O(n \times p)$ comparaison au total.
7. ```
   def chaine_mini(t):
       pos = 0
       for i in range(len(t)):
           if t[i][1] <= t[pos][1] :
               pos = 1
       return t[pos][0]
   ```
8. ```
   def tte_chaine_mini(t):
       L=[]  # contiendra les chaines minimales
       pos = 0
       for i in range(len(t)):
           if t[i][1] < t[pos][1] : # un plus petit
               pos = i
               L = [t[pos][0]] # on reinit L
           if t[i][1] == t[pos][1] : # un meme min
               L.append(t[i][0]) # on ajoute  L
       return L
   ```
9. ```
   Bilan = [] # contiendra des couples (nom,somme)
   for couple in Notes :
       nom,tab = couple
   ```
   ```

```
    #calul de la somme
    S = 0
    for val in tab:
        S+=val
    Bilan.append( (nom,S))

Boulet = tte_chaine_mini(Bilan)
```

EXERCICE 2 - LE TIPE DES VAINQUEURS

1. Voir le cours : se rappeler/retrouver que $u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}$ et

```
def newton(f,fp,x0,eps):
    u=x0
    v=u-f(u)/fp(u)
    while abs(v-u) > eps:
        u,v=v, v - f(v)/fp(v)
    return(v)
```

2. On a :
$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f^{(3)}(x) + o(h^3)$$
et
$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f^{(3)}(x) + o(h^3)$$
donc :
$$f(x+h) - f(x-h) = 2hf'(x) + 2\frac{h^3}{6}f^{(3)}(x) + o(h^3)$$
soit
$$\frac{f(x+h)-f(x-h)}{2h} - f'(x) = \frac{h^2}{6}f^{(3)}(x) + o(h^2) \sim \frac{h^2}{6}f^{(3)}(x)$$
3. Les données prendraient : $64 \times 1000 \times 60 \times 10 = 38400000$ bits, donc $38400000/(8 \times 10^6) = 4,8$Mo. La clé n'est pas assez grande.
4. Attention, on ne dispose pas de la fonction mais d'une liste des valeurs.

```
n = len(mes)
h = 0.001
S = 0
for i in range(1,n):
    S += h*(Lmes[i-1] + L[i])/2
print(S)
```

5. On a $Y' = (m', m") = (m', m^2 - tm) = F(t, Y)$ pour $F(t, a, b) = (b, a^2 - tb)$ où $F : \mathbb{R}^3 \to \mathbb{R}^2$.
6.
```
def F(t,Y):
     """ Fonction definssant l'equation differentielle"""
     m,mp = X
     return np.array([mp,m**2-t*mp])


...
# Euler
#for i in range(N):
for k in range(N) :
    Y = Y + h*f(t,Y)
    t = t + h
    tt.append(t)
    YY.append(Y)
```
7.
```
import matplotlib.pyplot as pl

YY= np.array(YY)
pl.plot([:,0],YY[:,1]))
pl.show()
```