

Correction

```
# Les premières questions n'ont que peu posé de problème
# Il faut néanmoins penser à les tester.
def entier(b,c):
    return 3*b+c

def poidsFort(x):
    if x<3:
        return 0
    else:
        return (x//3)

def poidsFaible(x):
    return x % 3

# Lecture de l'énoncé : si la longueur trouvée est supérieure à m,
# on considère que la chaîne la plus longue possible est de longueur m
def longueurMotif(t,i,j,m):
    longueur = 0
    while(i<len(t) and j<len(t) and t[i]==t[j]):
        longueur+=1
        i+=1
        j+=1
    return min(longueur,m-1)
# Linéaire, puisque demande au plus un passage le long de t, de taille N.

def longueurMotifMax(t,i,j,m):
    longueur=0
    k=0
    while (k<m and i+k<j):
        longueur = max(longueur, longueurMotif(t,i+k,j,m))
        k+=1
    return longueur
# quadratique puisqu'on fait un nombre de fois au plus linéaire
# un appel à une fonction linéaire

def motifMax(t,i,j,m):
    global A,L,C
    L=0
    k=0
    A = k
    while (k<m and i+k<j and j<len(t)):
        nLongueur = longueurMotif(t,i+k,j,m)
        # Lorsqu'on trouve mieux, on remplace les valeurs de A et L
        if L<nLongueur:
            A = k
            L = nLongueur
        k+=1
```

```
C = t[j+L]
return (A,L,C)
```

```
def imprimerTriplet(A,L,C):
    print(poidsFort(A),end='')
    print(poidsFaible(A),end="")
    print(poidsFort(L),end="")
    print(poidsFaible(L),end="")
    print(C,end="")
```

```
def compresser(t):
    j=9
    L = 0
    C = t[0]
    liste = []
    while (C<3):#(j+9<len(t)):
        lMax = len(t)-j
        (A,L,C) = motifMax(t,j-9,j,9)
        liste.append((A,L,C))
        imprimerTriplet(A,L,C)
        j=j+L+1
    #l'énoncé ne demande pas de renvoyer la liste des triplets, seulement
    #de les imprimer.
```

#chaque triplet donne une indication : on va chercher dans la fenêtre glissante
#la chaîne de longueur L à l'indice A (donc il faut garder en mémoire l'indice
#de début de la fenêtre), puis on rajoute le caractère C en fin de la liste
#qu'on est en train de construire.

```
def decompresser(tc):
    #liste est le texte décompressé
    liste = [0 for i in range(9)]
    indice = 0 #début de fenêtre
    # on lit le texte tc par blocs de 5 caractères
    for j in range (len(tc)//5):
        #traduction du texte vers A,L,C
        # le texte est une chaîne, on convertit les caractères en entiers
        A = entier(int(tc[j*5]),int(tc[j*5+1]))
        # print(tc[j*5],tc[j*5+1])
        L = entier(int(tc[j*5+2]),int(tc[j*5+3]))
        C = int(tc[5*j+4])
        # la chaîne commençant en A, de longueur L est recopiée en fin de liste
        #On recopie à la fin de la chaîne ce qui se trouve à l'indice "indice+ A"
        #de longueur L
        for k in range(L):
            liste.append(liste[indice+A+k])
        liste.append(C)
        #la fenêtre se décale de L+1
        indice+=L+1
    #mise à jour de l'indice de début de fenêtre
```

```
# On imprime enfin en sortie le texte décompressé
for e in liste:
    print(e,end='')
# return liste
```