

Question. On commence par donner deux fonctions :

```
def num(c):  
    return ord(c)-ord('A')  
def carac(n):  
    return chr(ord('A')+n)
```

La fonction **num** associe à chaque caractère son numéro : **num('A')** renvoie 0, **num('B')** renvoie 1, etc. **num('Z')** renvoie 25. L'autre fonction permet d'obtenir un caractère à partir de son numéro : **carac(0)** renvoie 'A', **carac(1)** renvoie 'B' etc.

Les messages considérés sont uniquement écrits avec les lettres majuscules 'A', 'B', ..., 'Z'.

Un message a été chiffré en utilisant un code à décalage, c'est à dire que chaque lettre du message de départ a été décalée suivant l'ordre de l'alphabet pour fournir la nouvelle lettre. Par exemple si on fait un décalage de 3, la lettre A est remplacée par D, B devient E, etc. On considère qu'après Z l'alphabet reprend dans l'ordre usuel donc Z décalée de 3 devient C. Mais en plus, dans le codage utilisé ici, le décalage change à chaque lettre du message de départ de la manière suivante : si la k -ième lettre du message de départ a été décalée de n_k , alors la suivante est décalée de n_{k+1} avec :

$$n_{k+1} = \begin{cases} \frac{n_k}{2} & \text{si } n_k \text{ est pair} \\ 3n_k + 1 & \text{sinon} \end{cases}$$

(on dit que (n_k) est une *suite de Syracuse*).

- (a) Coder le texte "BONJOUR" suivant la méthode donnée ci-dessus en prenant un décalage de 3 pour la première lettre.
- (b) Saurez-vous décoder le message "KRAWTVZGBAAWMKELRGWYUU" ?

Réponse. Une première fonction pour décaler une lettre d'un certain nombre k :

```
def decaler(c,k):  
    x = num(c)  
    y = (x+k)%26  
    return carac(y)
```

On note que si k est négatif, ceci réalisera le décodage. Par exemple si un texte a été codé en décalant de 3, on le décode en décalant de -3 . Le **%26** sert à remettre le nombre y entre 0 et 25. Ensuite une fonction pour coder toute une chaîne de caractères. On donne le décalage n_0 pour le premier caractère et les décalages suivants sont obtenus en utilisant la relation de récurrence de l'énoncé :

```
def coder(s,n0):  
    t = ""  
    n = n0  
    for c in s:  
        t = t+decaler(c,n)  
        if n%2==0:  
            n = n//2  
        else:  
            n = 3*n+1  
    return t
```

Et la même fonction pour décoder :

```
def decoder(s,n0):
    t = ""
    n = n0
    for c in s:
        t = t+decaler(c,-n)
        if n%2==0:
            n = n//2
        else:
            n = 3*n+1
    return t
```

```
print(coder("BONJOUR",3))
```

EYSZWYT

Pour décoder le message :

```
s = "KRAWTVZGBAAWMKELRGWYUU"
```

on écrit `print(decoder(s,k))` pour différentes valeurs de k à l'aide d'une boucle. On finit par trouver :

```
print(decoder(s,98))
```

QUIAINVENTELETELESCOPE

Remarque : on ne peut pas se contenter de faire `decoder(s,k)` avec $0 \leq k \leq 25$ car les suites obtenues en prenant comme valeur de départ k et $k+26$ ne restent pas forcément égales à 26 près au cours du décodage.