

**Exercice 2:** suite de Conway ( 1°) : 1 pt – 2°) : 2 pts – 3°) : 4 pts – 4°) : 2 pts )

L'objectif de cet exercice est de générer la suite mathématique de Conway . Cette suite se construit de la manière suivante : le premier terme de la suite de Conway est posé comme égal à 1. Chaque terme de la suite se construit en énonçant le terme précédent, c'est-à-dire en indiquant, en partant de la gauche, combien de fois chacun de ses chiffres se répète consécutivement.

L'exemple ci-après donne les premiers termes de cette suite avec leur expression littérale :

1	: premier terme de la suite qui s'énonce « un 1 » et produit ainsi le terme suivant 11
11	: 2ème terme de la suite qui s'énonce « deux 1 » et produit le terme suivant 21
21	: 3ème terme qui s'énonce « un 2 un 1 » et produit le terme suivant 1211
1211	: 4ème terme qui s'énonce « un 1 un 2 deux 1 » et produit le terme suivant 111221
111221	: 5ème terme qui s'énonce « trois 1 deux 2 un 1 » et produit le terme suivant 312211
312211	: etc...

On notera que le nombre maximum de répétitions d'un chiffre est toujours inférieur à 4, ce qui signifie qu'aucun terme de cette suite ne comporte un chiffre supérieur à 3.

Les termes seront représentés à l'aide de chaînes de caractères.

1) Écrire la fonction  $toChar(pDigit) \rightarrow car$   
qui à la valeur entière  $pDigit$  représentant un chiffre entre 0 et 9 retourne le caractère correspondant.

Exemple : `toChar(7)` retourne le caractère '7'

2) Écrire la fonction *nbApparition*(pTerme, pPos)  $\rightarrow$  entier  
avec

pTerme: **chaîne** : le terme qui est en cours de traitement

pPos : **entier** : la position du caractère courant dans *pTerme* pour lequel on va déterminer le nombre de répétitions consécutives.

Cette fonction détermine le nombre de répétitions consécutives du chiffre situé en position *pPos* dans le terme courant *pTerme*.

Exemple : avec *pTerme* = "1**222**1213" et *pPos* = 1, la fonction doit retourner la valeur 3.

3) *Écrire la fonction suivant*  $\text{pTerme} \rightarrow \text{chaine}$  :  
avec pTerme: **chaîne** : un terme de la suite de Conway

qui renvoie le terme suivant de la suite en appliquant le procédé décrit. Cette fonction devra **obligatoirement** appeler les fonctions *nbApparition* et *toChar* des questions précédentes, **même si vous ne les avez pas écrites.**

4) Écrire la procédure *generer*(pNb) :

qui affiche les *pNb* premiers termes de la suite de Conway. Cette procédure devra obligatoirement appeler la fonction *suivant* de la question précédente, même si vous ne l'avez pas écrite.