

TD14 : éléments de correction

Exercice 1 : Suite de Conway

Fonction : **ChiffreEnCaractere**

Entrée : - n : entier

Local : /

Sortie : - caractère

Début

Selon Cas n Faire

Cas 0 → Retourner '0'

Cas 1 → Retourner '1'

Cas 2 → Retourner '2'

Cas 3 → Retourner '3'

Cas 4 → Retourner '4'

Cas 5 → Retourner '5'

Cas 6 → Retourner '6'

Cas 7 → Retourner '7'

Cas 8 → Retourner '8'

Cas 9 → Retourner '9'

Sinon → **Exception** (valeur de n incorrecte)

Fin Selon Cas

Fin

Fonction : **NombreApparitionsConsecutives**

Entrée : - chaîne : chaîne de caractères

- terme : caractère

- indice : entier

Local : - i : entier

Sortie : - app : entier

Début

Soit app = 0

Soit i = indice

Tant Que i < chaîne.Longueur() - 1 ET chaîne[i] == terme **Faire**

Soit app = app + 1

Soit i = i + 1

Fin Pour

Retourner app

Fin

Fonction : **TermeSuivantConway**

Entrée : - termeCourant : chaîne de caractères

Local : - nbApparitions : entier

- nbApparitionsCaractere : caractère

- chiffreCaractere : caractère

- indice : entier

Sortie : - termeSuivant : chaîne de caractères

Début

Soit termeSuivant = ""

Soit indice = 0

Tant Que indice < termeCourant.Longueur() **Faire**

Soit nbApparitions = **NombreApparitionsConsecutives**(
termeCourant, termeCourant[indice], indice)

Soit nbApparitionsCaractere = **ChiffreEnCaractere**(nbApparitions)

Soit chiffreCaractere = **ChiffreEnCaractere**(**ASCII**(termeCourant[indice]) - **ASCII**('0'))

Soit termeSuivant = termeSuivant + nbApparitionsCaractere + chiffreCaractere

Soit indice = indice + nbApparitions

Fin Tant Que

Fin

Note : **ASCII**(caractère c) retourne le code ASCII donc un entier

En C# on pourra écrire : (int) termeCourant[indice] - (int) '0'

Fonction : GenererSuiteConway

Entrée : - n : entier

Local : - terme : caractère
- i : entier

Sortie : void

Début

```
| Si n < 0 Alors  
| | Exception (valeur de n invalide)  
| Fin Si  
| Soit terme = ""  
| Afficher("Conway(0) = {terme}")  
| Pour i de 1 à n avec un pas de +1 Faire  
| | Soit terme = TermeSuivantConway(terme)  
| | Afficher("Conway({i}) = {terme}")  
| Fin Pour  
Fin
```

Procédure : Main

Entrée : void

Local : - n : entier

Sortie : void

Début

```
| Afficher("Entrer n : ")  
| Soit n = LireEntier()  
| GenererSuiteConway(n)  
Fin
```