TP : Préparation à l'épreuve d'algo – BTS SIO

La suite de Syracuse

Une suite de Syracuse est une suite d'entiers naturels définie de la manière suivante : on part d'un nombre entier strictement positif ; s'il est pair, on le divise par 2 ; s'il est impair, on le multiplie par 3 et l'on ajoute 1. En répétant l'opération, on obtient une suite d'entiers strictement positifs dont chacun ne dépend que de son prédécesseur.

Ainsi, si on part de 14 :
14 est pair, donc on le divise par 2 et on obtient 7.
7 est impair, on le multiplie par 3 et on ajoute 1 : $7 \times 3 + 1 = 22$.
22 est pair, donc on le divise par 2 et on obtient 11.
11 est impair, on le multiplie par 3 et on ajoute 1 : $11 \times 3 + 1 = 34$.
34 est pair, donc on le divise par 2 et on obtient 17.
17 est impair, on le multiplie par 3 et on ajoute 1 : $17 \times 3 + 1 = 52$.
52 est pair, donc on le divise par 2 et on obtient 26.
26 est pair, donc on le divise par 2 et on obtient 13.
13 est impair, on le multiplie par 3 et on ajoute 1 : $13 \times 3 + 1 = 40$.
40 est pair, donc on le divise par 2 et on obtient 20.
20 est pair, donc on le divise par 2 et on obtient 10.
10 est pair, donc on le divise par 2 et on obtient 5.
5 est impair, on le multiplie par 3 et on ajoute 1 : $5 \times 3 + 1 = 16$.
16 est pair, donc on le divise par 2 et on obtient 8.
8 est pair, donc on le divise par 2 et on obtient 4.
4 est pair, donc on le divise par 2 et on obtient 2.
2 est pair, donc on le divise par 2 et on obtient 1.
1 est impair, on le multiplie par 3 et on ajoute 1 : $1 \times 3 + 1 = 4$.

On voit qu'à partir du moment où on obtient 1, on se trouve dans un cycle 1 – 4 – 2 – 1 - …

On définit le **temps de vol** comme le nombre d'itération avant d'obtenir 1.
Ainsi, pour 14, le temps de vol est de 17.

On définit l'**altitude maximale** comme la valeur maximale de la suite de Syracuse. Pour 14, l'altitude maximale est de 52.

Le but du sujet est de déterminer le plus petit entier de départ N nécessitant 20 étapes pour arriver à 1 (donc un temps de vol de 20) en précisant le plus grand nombre alors rencontré.

Partie A sur feuille :

1. Compléter l'écriture de la fonction `suivant(n)` ci-dessous afin qu'elle renvoie un nombre suivant `n`. **Par exemple**, on aura **`suivant`**`(17)=52`, `suivant(52)=26` et **`suivant`**`(1)=1`.

```
Fonction suivant(n: type entier)      # type du résultat : entier
Début
      Si n== ...  Alors
            Retourner(1)
      Sinon
            Si n%2== 0
                  Retourner(.........)
            Sinon
            Retourner(..........)
            FinSi
      FinSi
FinFonction
```

2. Quel est le rôle de l'algorithme suivant :

```
Variables : n (entier), ch (chaîne)
Début
      n←  14
      liste= []   # liste vide
      TantQue suivant(n) != 1 Faire
            liste ← liste + [suivant(n)]
            n ← suivant(n)
      FinTantQue
      Afficher liste + [1]
Fin
```

3. Écrire une fonction **trajet(n)** qui retourne la liste de tous les nombres rencontrés lorsqu'on part de l'entier n.
Par exemple on devra avoir :
`trajet(14)=[7,22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1]`.

4. Écrire une fonction **MAX(n)** qui à un nombre de départ n, retourne le plus grand nombre alors rencontré.

5. Écrire un algorithme déterminant puis affichant le plus petit entier N nécessitant 20 étapes pour arriver à 1 ainsi que le plus grand nombre alors rencontré.

6. Bonus. Écrire une fonction `syracuse(n)` qui retournera le plus petit entier N nécessitant n étapes pour atteindre 1, ainsi que le plus grand nombre rencontré.

Partie B sur machine :

1. Implémenter l'algorithme de la partie A. 5 en corrigeant les éventuelles erreurs.

2. Exécuter le programme puis noter les résultats obtenus.