

LES BOUCLES

Dans un algorithme, il peut arriver que l'on ait à répéter un certain nombre de fois une ou plusieurs instructions. On parle alors de boucle ou de structure itératives. Si le nombre de répétitions est connu par avance, on utilisera la structure FOR. En revanche, si ce nombre n'est pas connu et dépend du traitement des instructions de la boucle, on utilisera une boucle TANT QUE.

I Boucle POUR (FOR)

```

Déclarer I du type nombre
Pour I allant de 1 à n
  début
    instructions 1
    instructions 1
    ...
  fin
.....instruction suivante

```

Les instructions contenues dans la boucle Pour seront exécutées n fois. (de 1 à n, il y a n entiers). A chaque passage, la valeur du "compteur" I augmente de 1. Dès que la valeur du compteur atteint n, les instructions sont exécutées une dernière fois et l'on passe à la suite de l'algorithme.

- EXERCICE 1** Écrire un algorithme qui affiche le message "Tu es extraordinaire" 10 fois de suite.
- EXERCICE 2** Écrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. (Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27).
- EXERCICE 3** Écrire un algorithme qui demande un entier, qui calcule la somme des entiers entre 1 et l'entier saisi. Affichage de la somme. Prolongement : Algorithme qui calcule la somme des carrés des entiers de 1 à n. (saisi)
- EXERCICE 4** On définit la suite (u_n) par : u_{n+2} = u_{n+1} + u_n et u₀ = 1 et u₁ = 1. (suite de Fibonacci) Écrire un algorithme qui demande un entier, qui calcule le terme de rang l'entier saisi. Affichage du terme.

II Boucle TANT QUE (WHILE)

```

Initialisation de la condition
Tant Que <condition> Faire
  début
    instructions 1
    instructions 1
    ...
  Recalcul de la condition
  ...
fin Tant que
.....instruction suivante

```

Le principe de la boucle est simple : le programme arrive sur la ligne du Tant Que. Il examine alors la condition. Si la condition est VRAIE, le programme exécute les instructions qui suivent, jusqu'à ce qu'il rencontre la ligne Fin Tant Que. Il retourne ensuite sur la ligne du Tant Que, procède au même examen, et ainsi de suite. Le manège enchanté ne s'arrête que lorsque la condition est FAUSSE.

- EXERCICE 5** Écrire un algorithme qui affiche le message "Tu es formidable" autant de fois que l'utilisateur le demande.
- EXERCICE 6** Un programme pose une question à laquelle l'utilisateur doit répondre par O (Oui) ou N (Non). Il se peut que l'utilisateur tape autre chose que la réponse attendue. Dès lors, le programme peut planter. Écrire un algorithme qui repose la même question jusqu'à l'une des deux réponses attendues soit saisie au clavier par l'utilisateur. Cela s'appelle un contrôle de saisie.
- EXERCICE 7** Écrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.