

TP N° 4 : Les structures répétitives

Schéma itératif TANTQUE

```
while < Expression booléenne > do  
  begin  
    < Instruction / bloc d'instructions >  
  end;
```

- Les variables de l'expression booléenne doivent être initialisées avant le **while**, pour que l'expression puisse être évaluée au premier passage.
- *while* boucle tant que l'expression booléenne est vrai. Pour éviter une boucle infinie, il faut obligatoirement que dans le bloc d'instructions il y ait une instruction permettant de passer aux itérations suivantes et rendant l'expression fausse à un moment donné.

Schéma itératif REPETER JUSQU'A

```
repeat  
  < Instruction / Bloc d'instructions >  
until  
  <Expression booléenne>
```

- Les instructions sont exécutées, puis l'expression booléenne est évaluée. Si l'expression est vraie, alors on s'arrête, sinon on boucle à nouveau.
- Pas besoin d'encadrer le groupe d'instructions par begin et end ; les mots clé **repeat** et **until** jouent déjà ce rôle.

Schéma itératif POUR

```
for i:=<valeur initiale> to <valeur finale> do  
  begin  
    < Instruction / Bloc d'instructions >  
  end
```

- i est le compteur de la boucle
- Valeur initiale et valeur finale sont les bornes inférieures et supérieures (du même type que i)
- Pour avoir une boucle décroissante, on écrit :

```
for i:=<valeur finale> downto <valeur initiale> do  
  begin  
    < Instruction / Bloc d'instructions >  
  end
```

Exercice 1

Ecrire un programme qui demande à l'utilisateur de saisir N notes, puis retourne le nombre de notes qui sont supérieures à 10. Pour terminer la saisie des notes, on fournira une valeur négative.

N.B : La note est comprise entre 0 et 20.

Exercice 2

Ecrire un programme qui permet de compter le nombre de chiffres d'un nombre et la somme de ses chiffres.

Exercice 3

Ecrire un programme qui permet de lire un nombre et d'inverser l'ordre de ses chiffres.

Exemple : N = 1978963 l'inverse est 4891

Exercice 4

Ecrire un programme qui demande à l'utilisateur d'entrer la valeur d'une variable réelle comprise entre deux bornes, un min et un max, et qui renvoie cette valeur. Le programme doit afficher un message d'erreur si la valeur saisie n'est pas comprise entre les bornes et redemander à l'utilisateur une nouvelle saisie jusqu'à pouvoir renvoyer une valeur correcte.

Exercice 5

Ecrire un programme qui demande successivement des nombres à l'utilisateur, et retourne ensuite le plus grand parmi ces nombres ainsi que sa position. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

Exercice 6

Ecrire un programme qui permet de convertir un entier N écrit sous forme binaire en sa valeur décimale.