

LECON 3 :

STRUCTURE ITERATIVES.



Enseignant : Mr AMZIL

CHAPITRE 3 : Les structures Itératives (Boucles).

La structure répétitive permet de répéter un ensemble d'instructions jusqu'à ce qu'une condition soit réalisée. il existe plusieurs formes de structures répétitives :

- la structure Tant que.... fin tant que
- La structure Répéter jusqu'à
- La structure Pour.... fin pour

I : LA BOUCLE TANT QUE...:

Lorsque la boucle tant que est rencontrée la condition d'entrée dans la boucle est évaluée.

Syntaxe :

```
Tant que <expression logique> faire  
  <instructions>  
Fin Tant que
```

Description :

On teste d'abord si <expression logique> est vraie. Dans ce cas, on exécute <instructions> puis on boucle de nouveau sur le test de <expression logique>

Par contre si <expression logique> est fausse, la boucle se termine et le programme poursuit son exécution après «fin tant que».

exemple :

Algo qui affiche dix fois bonjour sur l'écran.

ALGO bonjour

Var k: entier

Debut

k<- 1

Tant que k<=10 faire

Afficher («bonjour»)

k<- k+1

Fin tant que

Fin

II : LA BOUCLE REPETER... JUSQU'A:

Syntaxe :

```
Répéter  
  <instructions>  
jusqu'à <expression logique>
```

Description :

Le bloc <instructions> est exécutée

l'<expression logique> est testée

Dans le cas ou elle est égale à faux, on recommence au point 1.

Dans le cas ou elle est égale à vrai, le programme poursuit son exécution après l'instruction «jusqu'à».

Exemple :

```
Algo table de multiplication de dix
ALGO table de 10
  Var k: entier
  Debut
  afficher(«Table de multiplication de 10»)
  k <- 1
  répéter
    Afficher k, «*»10 «=» k*10
    k <- k+1
  jusqu'à k > 10
Fin
```

La structure itérative répéter jusqu'à s'exécute au moins une fois car le test(l'expression logique) se trouve en fin de structure contrairement au tant que.

III : LA BOUCLE POUR... FINPOUR:

Syntaxe :

Pour <indicateur> de <expression début> à <expression fin> [pas de<expression pas>] Faire
<instructions>
Fin pour

Description :

<indicateur> est une variable assimilée à un compteur qui est automatiquement augmenté ou diminué en fonction de <expression pas>.

Au début, <indicateur> prend la valeur de <expression début>

Le bloc d'instruction est exécuté jusqu'à ce que <indicateur> prenne la valeur de <expression fin>

la valeur <indicateur> évolue de <expression début> jusqu'à <expression fin> en fonction de <l'expression pas> qui par défaut est égale à 1.

Exemple :

```
Algo table de multiplication de dix
ALGO table de 10
  Var k: entier
  Debut
  k 1
  afficher («table de multiplication de 10»)
  Pour k de 1 à 10 faire
    afficher k «*10 =» k*10
  fin pour
Fin
```

Dans le Pour, il n'y a pas de compteur.

Dans cet algorithme, le pas de progression de l'indice k est implicitement 1.

x 0 (initialisation) repartir à 0.

xx+2 (incrément) ajouter une valeur positif.

xx-2 (décrément) enlever une valeur.

Exercice n°1 :

Table de multiplication.

Ecrire un algorithme qui demande de saisir un nombre compris entre 1 et 10 puis affiche sa table de multiplication. (utiliser une des boucles).

Exercice 2:

Saisir le nom et prénom de l'élève plus la note obtenue au bac selon la note afficher la mention.

Exercice 3 :

Ecrire un algorithme qui demande à l'utilisateur de saisir une suite de nombre et d'afficher la somme au fur et à mesure. Demander a chaque saisie si l'utilisateur veut continuer (0 pour oui et 1 pour non).