

### 3.3. Structures itératives (boucles) :

Définition :

- \* Une boucle c'est l'exécution d'une même partie de programme plusieurs fois.
  - \* Une boucle doit toujours s'exécuter un nombre fini de fois (itération)
  - \* Itération : répétition d'un même traitement plusieurs fois.
- Faire 5 itérations  $\Rightarrow$  passer 5 fois dans la boucle.

#### Boucle à bornes définies : la boucle POUR

Notation

POUR variable VARIANT DE valeur initiale A valeur finale FAIRE  
<action>

Principe : on utilise une variable qui sert de compteur. On connaît le nombre d'itérations avant d'entrer dans la boucle.

Quand utiliser la boucle POUR : Lorsqu'on connaît, avant de rentrer dans la boucle, le nombre d'itérations.

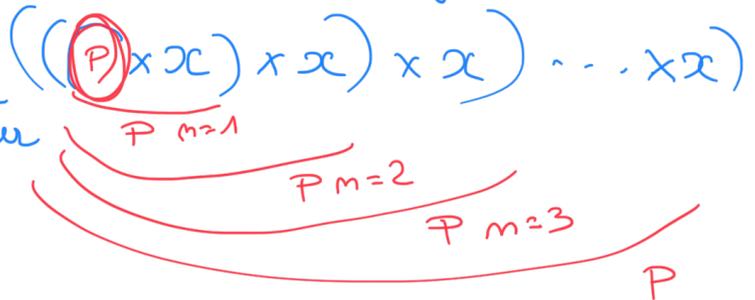
Exemple : Calcul de  $x^n$ .

$n > 0$

$x \times x \times x \times \dots \times x$   
n fois

données  $x$  réel  
 $n$  entier

variable  $P$  réel : résultat  
 $k$  entier compteur



hypothèse  $n \geq 0$

$P \leftarrow 1$

pour  $k$  allant de 1 à  $n$  faire

$P \leftarrow P \times x$

~~$x$~~  : 2

~~$n$~~  : 4

~~$P$~~  : ~~1~~ ~~2~~ ~~4~~ ~~8~~ 16

~~$k$~~  : ~~1~~ ~~2~~ ~~3~~ 4

~~pour  $k$  allant de 1 à  $n$  faire~~

~~$x \leftarrow x \times x$~~

~~faux.~~  
 ~~$x$  est modifié à chaque pas de boucle~~

~~$x$~~  : ~~2~~ ~~4~~ ~~16~~ 256  
 ~~$n$~~  : 4 256 256

~~$k$~~  : ~~1~~ ~~2~~ ~~3~~ 4

hypothèse  $n > 0$

$P \leftarrow x$

pour  $k$  allant de 2 à  $n$  faire

$P \leftarrow P \times x$  donnée

## Boucles à bornes non définies : la boucle TANT QUE

Notation

TANT QUE <condition> FAIRE <actions>

*la condition*

Principe : Tant que l'expression est vraie on effectue les actions de la boucle.

\* C'est une répétition conditionnelle

\* Les actions de la boucle doivent faire évoluer les variables testées dans <condition> sinon on boucle indéfiniment.

\* Si la condition est fautive dès l'entrée, on ne rentre pas dans la boucle.

\* La condition doit être connue avant l'évaluation de l'expression.

*doit pouvoir être évaluée dès l'entrée dans l'instruction*

Quand utiliser la boucle tant que : Lorsqu'on ne connaît pas le nombre d'itérations.

Remarque : toute forme itérative peut s'écrire sous la forme d'une boucle TANTQUE

Exemple : Calculer le PGCD de deux nombres sans utiliser Modulo et Division euclidienne

18	6	R = 18 - 6 = 12
12	6	R = 12 - 6 = 6
6	6	PGCD : 6

2 entiers positifs

$a = 18$   
 $b = 6$   
 $18 - 6 = 12$   
 $12 - 6 = 6$   
 PGCD = 6.

$a = 12$   
 $b = 8$   
 $12 - 8 = 4$   
 $8 - 4 = 4$   
 PGCD : 4

données a, b entiers

variable pgcd entier résultat

x, y entier, pour ne pas modifier les données

hypothèse a, b > 0

actions

$x \leftarrow a$   
 $y \leftarrow b$

initialisation  
 condition

tant que  $x \neq y$  faire

action

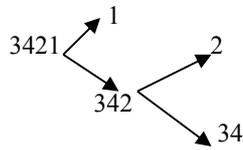
si  $x > y$   
 alors  $x \leftarrow x - y$   
 sinon  $y \leftarrow y - x$

a : 12  
 b : 8  
~~x : 12~~ 4  
~~y : 8~~ 4  
 PGCD : 4

PGCD  $\leftarrow x$  (ou PGCD  $\leftarrow y$ )

Exemple : calculer la valeur décimale d'un nombre octal

$$(3421)_8 \rightarrow 1 * 8^0 + 2 * 8^1 + 4 * 8^2 + 3 * 8^3 \rightarrow (1816)_{10}$$



Exemple : Calcul de  $x^n$ .

données  $m$  : entier exposant  
 $x$  : réel

variables  $P$  : réel, résultat  
 $k$  : entier, compteur de boucle

hypothèses :  $m \geq 0$

actions :

$P \leftarrow 1$   
 $k \leftarrow 0$

$x^0$        $x^k, k=0$

initialisation  
 des variables  
 de boucle

$m=0$   
 $x=2$

$P: 1 = 2^0$   
 $k: 0$

tant que  $k \neq m$  faire

condition

$P \leftarrow P \times x$   
 $k \leftarrow k + 1$

actions  
 de boucle

$P \leftarrow x$   
 $k \leftarrow 1$

tant que  $k \neq m$  faire

$P \leftarrow P \times x$   
 $k \leftarrow k + 1$

hypothèse  
 $m > 0$

$$m = 4$$

$$x = 2$$

$$P: \cancel{2} \cancel{4} \cancel{8} \textcircled{16}$$

$$R: \cancel{1} \cancel{2} \cancel{3} 4$$

### Boucles à bornes non définies : la boucle REPETE JUSQU'A

#### Notation

REPETER <actions> JUSQU'A <expression>.

On répète les actions autant de fois que l'expression est fausse (la condition d'arrêt est inverse de la Boucle TANT QUE)

- \* c'est aussi une répétition conditionnelle
- \* il faut aussi que la condition varie dans le bloc d'instructions sinon on boucle indéfiniment
- \* il y a au moins un passage dans la séquence d'instructions quelle que soit la condition d'arrêt.

Quand utilise t-on le REPETER : Lorsque l'on ne connaît pas le nombre d'itérations de la boucle et que l'on doit effectuer les actions au moins une fois.

Exemple : Calculer le PGCD de deux nombres sans utiliser Modulo et Division euclidienne

données  $a, b$  entiers  
 variables  $x, y$  entiers  
 pgcd entier  
 hypothèse  $a, b > 0, \underline{a \neq b}$   
 actions

$x \leftarrow a$   
 $y \leftarrow b$

initialisation

repete

si  $x > y$   
 alors  $x \leftarrow x - y$   
 sinon  $y \leftarrow y - x$

actions boucle

jusqu'à  $\textcircled{x = y}$  condition

pgcd  $\leftarrow x$

a: 4  
 b: 4  
~~x: 4~~ 4  
~~y: 4~~ 0  
 PGCD

#### 3.4. Le bloc ou la séquence d'action

Pour éviter les ambiguïtés de lecture d'un algorithme, une séquence d'action peut être mise entre accolades

<actions> peut s'écrire

```
{
  Action1
  Action2
  ...
  ActionN
}
```

tant que  $k \neq n$  faire

```
{
  P ← P * x
  R ← R + 1
}
```

### Exemple

Données    n : entier, exposant  
          x : nb à valeur réelle  
Variables  P : résultat, réel  
          k : indice de boucle, entier

Hypothèse  $n \geq 0$

Actions

```
{     P ← 1  
      k ← 0  
      TANTQUE k ≠ n FAIRE  
          { P ← P * x  
            k ← k + 1  
          }  
}
```

### 4. Remarques

On peut rajouter une action qui indique que la valeur d'une donnée est entrée par l'utilisateur. Cette action peut s'écrire

Lire (A) : demande à l'utilisateur une valeur pour A et affecte cette valeur à la donnée A

De même, pour indiquer un résultat à l'utilisateur, on peut utiliser une action dite d'affichage. Cette action peut s'écrire :

Affiche (A) : affiche la valeur de A.

#### Exemple 1 :

Données    n : entier, exposant  
          x : nb à valeur réelle  
Variables  P : résultat, réel  
          k : indice de boucle, entier

Hypothèse  $n \geq 0$

Actions

```
{     Lire(n) , Lire(x)  
      P ← 1  
      k ← 0  
      TANTQUE k ≠ n FAIRE  
          { P ← P * x  
            k ← k + 1  
          }  
      Affiche(P)  
}
```

Exemple 2 : Calcul de la moyenne d'une suite de nombres.

```
Données Nb : nombre de nombres entiers
        X : nb entier (prend des valeurs successives)
Variables i : indice de boucle entier
        Somme, Moyenne : entier et réel.
Actions
{
  Somme ← 0
  Lire (Nb)
  POUR i VARIANT DE 1 à Nb FAIRE
  {
    Lire (X)
    Somme ← Somme + X
  }
  Moyenne ← Somme / Nb
  Afficher (Moyenne)
}
```