

Chapitre 4:

Les structures répétitives (Les boucles)

(en algorithmique et en langage C)

Instructions à structure répétitive

Problème :

Écrire un algorithme permettant d'afficher 300 fois le message :
“ *bonjour tout le monde*”.

Solution Classique :

DEBUT

Écrire (' Bonjour tout le monde ') **1**

Écrire (' Bonjour tout le monde ') **2**

.....

.....

Écrire (' Bonjour tout le monde ') **300**

FIN

Instructions à structure répétitive

Lorsqu'on a besoin de répéter une action un certain nombre de fois, on utilise les structures répétitives. La plupart des langages de programmation proposent trois types des boucles.

On distingue trois sortes de boucles en langages de programmation :

- Les boucles **tant que** : on y répète des instructions tant qu'une certaine condition est réalisée
- Les boucles **répéter jusqu'à** : on y répète des instructions jusqu'à ce qu'une certaine condition soit réalisée
- Les boucles **pour** : on y répète des instructions en faisant évoluer un compteur (variable particulière) entre une valeur initiale et une valeur finale

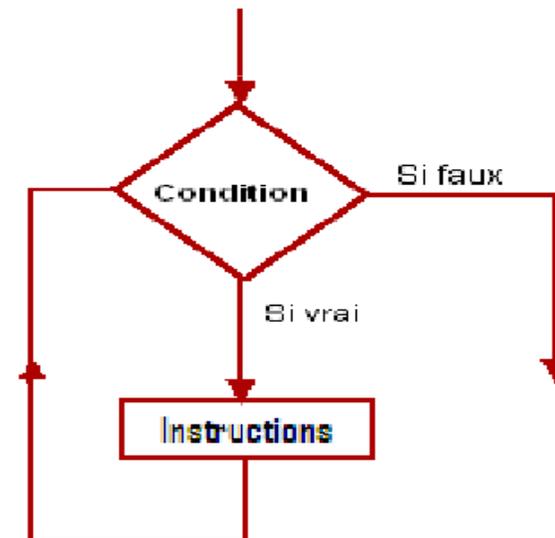
Instructions à structure répétitive

1. La boucle TantQue

On utilise cette instruction pour exécuter des actions tant qu'une condition est réalisée.

Syntaxe :

```
<Initialisation>  
TantQue Condition Faire  
    < Instructions >  
Fintantque
```



Tant que la condition est vraie, on exécute les instructions.

Instructions à structure répétitive

La boucle TantQue

Solution de l'exercice précédent :

Variable **i : Entier**

DEBUT

i ← 0 * *Initialisation* *

Tantque (i < 300) **Faire**

Écrire (' Bonjour tout le monde ')

i ← i + 1 * *incrémentation du compteur* *

Fintantque

FIN

Instructions à structure répétitive

La boucle TantQue

Exercice : Afficher tous les multiples de 9 inférieurs à 1000

Algorithme Multiples de 9

Variable M, i : **Entier** **i: variable intermédiaire: compteur**

DEBUT

 i ← 0 ** initialisation du compteur**

 M ← 0

Tantque (M < 1000) **Faire**

Écrire (M, ' est un multiple de 9 ')

 i ← i + 1 ** incrémentation du compteur **

 M ← i * 9

Fintantque

FIN

Instructions à structure répétitive

La boucle TantQue

Dans cet algorithme combien de fois la boucle est- elle exécutée ?

Algorithme Boucle1

Variable i , y : Entier

Début

i ← 2

y ← 0

TantQue (i<7) **faire**

 i ← i+1

 y ← y+i

Écrire (' Y = ' , y)

FinTantque

Écrire ('I=', i, 'Y=',y)

Écrire ('Fin Algorithme')

Fin

i = variable intermédiaire = **compteur**
i ← i+1 = **incréméntation** du compteur

Exécution :

Y= 3

Y= 7

Y=12

Y=18

Y=25

I=7 Y=25

Fin Algorithme

Instructions à structure répétitive

La boucle TantQue

Dans cet algorithme combien de fois la boucle est- elle exécutée ?

Algorithme Boucle2

Variable n : Entier

Début

n ← 1

TantQue (n>0) **faire**

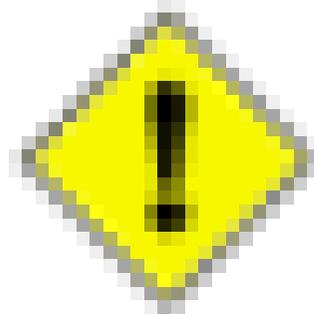
 Écrire ('N=', n)

 n ← n+2

FinTantque

Écrire ('Fin Algorithme')

Fin



Attention
Boucle infinie

Exécution :

N=1

N=3

N=5

N=7

N=9

N=11

N=13

.....

Instructions à structure répétitive

La boucle TantQue

Exercice:

Calculez la somme d'une suite de chiffres non nuls entrés au clavier, sachant que la suite est terminée par zéro.

Instructions à structure répétitive

La boucle TantQue

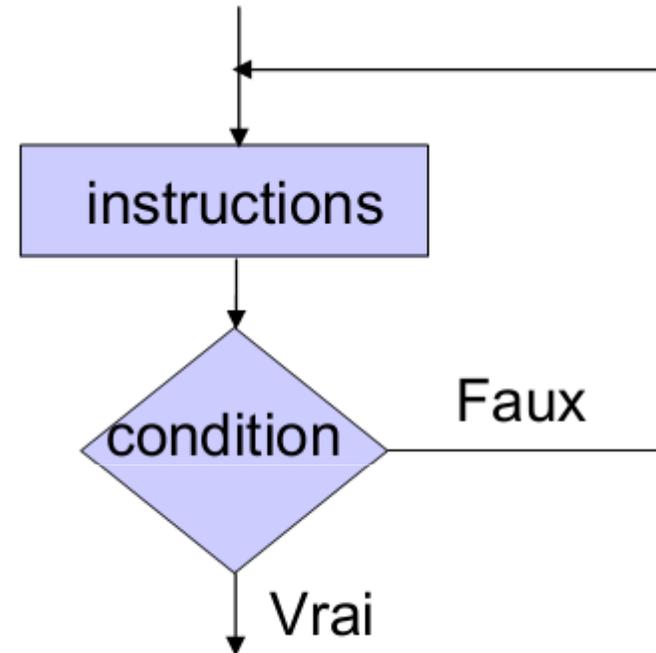
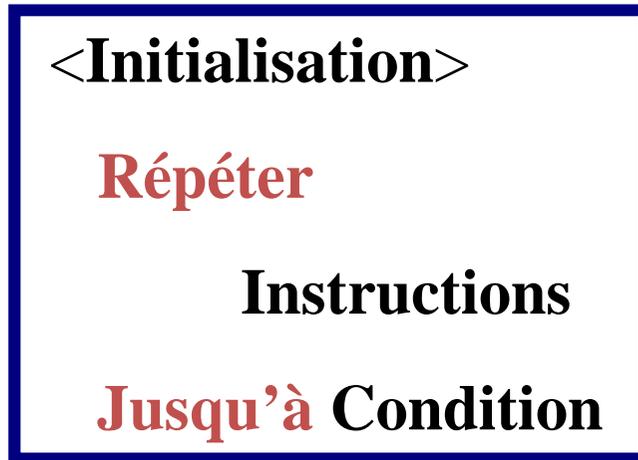
Solution :

```
Algorithme SOMME
Variables : Som, Nombre : Entier
DEBUT
  Som ← 0
  ECRIRE ("donner un nombre")
  LIRE (Nombre)
  Tantque Nombre <> 0 faire
    Som ← Som + Nombre
    ECRIRE ("donner un nombre")
    LIRE (Nombre)
  FinTantQue
  ECRIRE ("La somme est : ", Som)
FIN
```

Instructions à structure répétitive

Boucle « Répéter jusqu'à »

Syntaxe :



Ce qui signifie que l'on exécute les **instructions** jusqu'à ce que la **condition** soit vraie.

Remarque :

Contrairement à la boucle tant que, les instructions seront exécutées au moins une fois.

Instructions à structure répétitive

Boucle « Répéter jusqu'à »

Exemple

Algorithme Boucle2

Variable i, y : Entier

Début

$i \leftarrow 2$

$y \leftarrow 0$

Répéter

$i \leftarrow i+1$

$y \leftarrow y+i$

Écrire (' y = ', y)

Jusqu'à ($i < 7$)

Fin

Valeurs de y ?

Instructions à structure répétitive

Boucle « Répéter jusqu'à »

Exemple

Algorithme Boucle2

Variable i, y : Entier

Début

$i \leftarrow 2$

$y \leftarrow 0$

Répéter

$i \leftarrow i+1$

$y \leftarrow y+i$

Écrire (' y = ', y)

Jusqu'à ($i = 7$)

Fin

Donner les valeurs de y

Instructions à structure répétitive

Boucle « Répéter jusqu'à »

Exercice :

Écrire un algorithme permettant de calculer, pour un entier $N > 0$, la somme :

$$S_N = 1 + 2 + 3 + \dots + N$$

Instructions à structure répétitive

Boucle « Répéter jusqu'à »

Solution :

Variable N, S, i : *Entier*

DEBUT

Écrire (' Saisir une valeur entière positive :')

Lire (N)

$S \leftarrow 0$ * *initialisation de la boucle**

$i \leftarrow 0$

Répéter

$i \leftarrow i + 1$

$S \leftarrow S + i$

jusqu'à ($i \geq N$)

Écrire (' La somme : S = ', S)

FIN

Instructions à structure répétitive

Boucle « Répéter jusqu'à »

Exercice:

Refaire le même exemple précédent avec la boucle Répéter....Jusqu'à

Calculez la somme d'une suite de chiffres non nuls entrés au clavier, sachant que la suite est terminée par zéro.

Instructions à structure répétitive

Boucle « Répéter jusqu'à »

Solution :

```
Algorithme   SOMME
Variables : Som, Nombre : Entiers
DEBUT
  Som ← 0
  Répéter
    ECRIRE ("donner un nombre")
    LIRE (Nombre)
    Som ← Som + Nombre
  Jusqu'à Nombre = 0
  ECRIRE ("La somme est : ", Som)
FIN
```

Instructions à structure répétitive

Boucle « boucle Pour.... finPour »

On utilise cette boucle lorsque le nombre d'itérations à exécuter est **connu avant**.

Syntaxe :

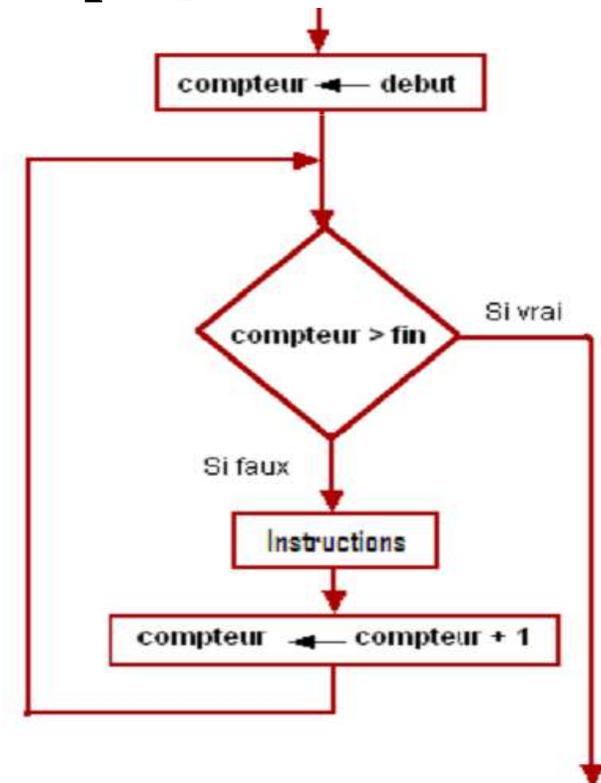
POUR compteur **DE** début **À** fin [**PAS** val_pas]

Instructions

FinPour

Remarque :

La valeur par défaut du PAS est égal à +1.



Instructions à structure répétitive

Boucle « boucle Pour.... finPour»

Exercice :

Écrire un algorithme permettant le calcul du factoriel d'un entier $N > 0$ donné : $N!$

Instructions à structure répétitive

Boucle « boucle Pour.... finPour»

Solution :

Algorithme Factoriel

Variable N : Entier

i : Entier

*** i variable intermédiaire**

F : Entier

i = compteur *

DEBUT

Écrire (' Saisir une valeur entière N > 0: ')

Lire (N)

F ← 1 * initialisation de la boucle*

Pour i de 1 à N

F ← F*i

Fin Pour

Écrire (' Le factoriel de ', N, ' est : ', F)

FIN

Instructions à structure répétitive

Boucle « boucle Pour.... finPour»

Exercice :

Écrire l'algorithme permettant de calculer la moyenne des notes de N étudiants

Instructions à structure répétitive

Boucle « boucle Pour.... finPour»

Solution :

Algorithme Moyenne

Variable *N, i : Entier*

note, S, Moy : Réel

DEBUT

Écrire (' Saisir le nombre d'étudiants: ')

Lire (N)

*S ← 0 * initialisation de la boucle**

Pour, i de 1 à N

Écrire (' Saisir la note de l'Etudiant ', i, ' :')

Lire (note)

S ← S + note

Fin Pour

Moy ← S/N

Écrire (' La moyenne est : ', Moy)

FIN

Instructions à structure répétitive

Boucle « boucle Pour.... finPour»

Exercice

Écrire un algorithme permettant d'afficher les lettres de l'alphabet.

Solution :

Algorithme alphabet

Variable lettre: **Caractère**

** parcourir les lettres de l'alphabet**

Début

Pour lettre **de** ' a ' **à** ' z '

Écrire(lettre)

FinPour

Fin

Instructions à structure répétitive

Quelle répétition ou Boucle choisir ?

Si nombre d'itérations connu **Alors**

Boucle Pour

Sinon

Si itération exécutée au moins une fois **Alors**

Boucle Répéter jusqu'à

Sinon

Boucle Tantque faire

Instructions à structure répétitive

Boucles « Tantque faire » et « Répéter jusqu'à »

Remarques: Soient **T** une condition et **R** l'action. Alors il y a **équivalence** entre les boucles **Tantque faire** et **répéter jusqu'à**.

La syntaxe est la suivante:

■ TantQue T faire
R
FinTantQue

~

■ Si T alors
Répéter
R
Jusqu 'à non(T)
FinSi

Et

■ Répéter
R
Jusqu 'à T

~

■ R
TantQue non(T) faire
R
FinTantQue

Instructions à structure répétitive

Lien entre Pour et TantQue

La boucle **Pour** est un cas particulier de **TantQue** (cas où le nombre d'itérations est connu et fixé) . Tout ce qu'on peut écrire avec **Pour** peut être remplacé avec **TantQue** (la réciproque est fausse)

Pour compteur **de** initiale **à** finale **pas** valeur du pas
instructions
FinPour

peut être remplacé par : (cas d'un pas positif)

```
compteur ← initiale
TantQue compteur <= finale faire
  instructions
  compteur ← compteur+pas
FinTantQue
```

Instructions à structure répétitive

Lien entre Pour et TantQue

Exemple:

Calcul de x à la puissance n où x est un réel non nul et n un entier positif ou nul (forme avec TantQue)

```
Variables x, puiss : réel
           n, i : entier
Début
Ecrire (" Entrez respectivement les valeurs de x et n ")
Lire (x, n)
puiss ← 1, i ← 1
TantQue (i≤n) faire
    puiss← puiss*x
    i ← i+1
FinTantQue
Ecrire (x, " à la puissance ", n, " est égal à ", puiss)
fin
```

```
puiss ← 1
Pour i de 1 à n
    puiss← puiss*x
finpour
```

Instructions à structure répétitive

Boucles imbriquées

Les instructions d'une boucle peuvent être des instructions itératives. Dans ce cas, on aboutit à des **boucles imbriquées**

Exemple:

Pour i de 1 à 5

Pour j de 1 à i

écrire("O")

FinPour

écrire("K")

FinPour

Exécution

OK

OOK

OOOK

OOOOK

OOOOOK

Instructions à structure répétitive

Choix d'un type de boucle

- Si on peut déterminer le nombre d'itérations avant l'exécution de la boucle, il est plus naturel d'utiliser la boucle **Pour**
- S'il n'est pas possible de connaître le nombre d'itérations avant l'exécution de la boucle, on fera appel à l'une des boucles **TantQue** ou **répéter jusqu'à**
- Pour le choix entre **TantQue** et jusqu'à :
 - ✓ Si on doit tester la condition de contrôle avant de commencer les instructions de la boucle, on utilisera **TantQue**
 - ✓ Si la valeur de la condition de contrôle dépend d'une première exécution des instructions de la boucle, on utilisera **répéter**

De l'algorithmique au C

Traduction des instructions : boucle Pour

Syntaxe en algo	Syntaxe en C
Pour i de «valeur initiale» à «valeur finale» [pas p] instructions Fin Pour	<pre>for (i = «valeur initiale» ; i <= «valeur finale» ; i = i + p) { instructions }</pre>
Pour i ← 1 à 100 de pas 3 Faire ... Fin Pour	<pre>for (i = 1 ; i <= 100 ; i = i + 3) { ... }</pre>
Pour i ← 1 à 100 Faire Fin Pour	<pre>for (i = 1 ; i <= 100 ; i++) { }</pre>

De l'algorithmique au C

Traduction des instructions : boucle Pour

Exemple : traduire en C l'algorithme suivant

Algorithme Factorielle

Variable n, fact : Entier

Début

fact ← 1

Pour n ← 1 à 100 Faire

fact ← (n * fact)

Fin pour

Ecrire ("La factorielle de 100 est ", fact)

Fin

Corrigé

```
#include <stdio.h>
Void main ( )
{
int n, fact;
fact = 1 ;
for (n=1; n<=100; n++)
{
fact = (n * fact) ;
}
printf ("La factorielle de 100 est %d", fact) ;
}
```

De l'algorithmique au C

Traduction des instructions : boucle Tantque et répéter jusqu'à

Syntaxe en algo	Syntaxe en C
Tantque condition Faire instructions FinTantque	while (condition) { instructions }
Répéter instructions Jusqu'à condition	do { instructions } while (condition) ;

De l'algorithmique au C

Exemple : traduire en C l'algorithme suivant

```
Algorithme Saisie
Variable n : Entier
Début
  Lire (n)
  Tantque n > 0 Faire
    Écrire ("Saisissez un nombre")
    Lire(n)
  FinTantque
Fin
```

Corrigé

```
#include <stdio.h>
Void main ( )
{
  int n ;
  scanf("%d",&n);
  while (n > 0)
  {
    printf("Saisissez un nombre");
    scanf("%d",&n);
  }
}
```