

CH3

Les Structures Répétitives

1^{ère} Année Licence GL

**Kais ben Salah
&
Taoufik Sakka Rouis**

Les structures répétitives

1- Introduction

Il est souvent nécessaire d'exécuter plusieurs fois une action ou un groupe d'actions, non pas indéfiniment mais un certain nombre de fois (pas forcément connu à l'avance) :

→ c'est la notion de boucle.

Les structures répétitives

2. La structure « for »

Syntaxe :

```
for(<expression_I>; <expression_C> ; <expression_M>)  
{  
    <Bloc d'instructions> ;  
}
```

- ❑ **expression-I** : une expression initialisant les variables de contrôle qu'il faut initialiser avant d'entrer dans la boucle.
- ❑ **expression-C** : la condition de bouclage.
- ❑ **expression-M** : une expression permettant de modifier les variables de contrôles utilisées.

Les structures répétitives

Déroulement de l'instruction "for"

1. Exécuter les instructions initiales de « for » (expression_I).
2. Évaluer la condition : (expression_C)
 - ☐ Faux (0) : branchement à l'instruction qui suit le fin du bloc « for ».
 - ☐ Vrai (1) :
 - ✓ Exécution les instructions du corps « for ».
 - ✓ Exécution de la liste des instructions de modification (expression_M).
 - ✓ Revenir à (2)
3. Fin de l'instruction « for ».

Les structures répétitives

Exemple 1 : /* Calcul de la somme : 1+2+3.....+10*/

Méthode 1

```
int i ,S;  
for( i=1 , S=0 ; i<=10 ;vi++)  
    S+=i ;  
printf("La somme =%d\n ",S);
```

Méthode 2

```
int i=1,S=0 ;  
for( ; i<=10 ; )  
{  
    S+=i ;  
    i++;  
}  
printf("La somme = %d\n ",S) ;
```

Méthode 3

```
int i=1,S=0 ;  
while(i<=10)  
{  
    S+=i ;  
    i++ ;  
}  
printf("La somme =%d\n ",S);
```

Méthode 4

```
int i , s;  
for(i=1, S=0 ; i<=10 ; S+=i, i++) ;  
printf("La somme = %d\n ",S) ;
```

Les structures répétitives

Exemple 2 :

Calculer et afficher la somme des diviseurs d'un entiers N sauf lui-même.

```
for (i=2 , S=1; i<=N/2 ; i++ )  
    if(N%i==0)  
        s+=i;
```

Les structures répétitives

3. La structure « while »

Syntaxe :

```
while(<expression>)  
{  
    < Bloc d'instructions> ;  
}
```

- ❑ Tant que l'<expression> fournit une valeur différente de zéro, le <bloc d'instructions> est exécuté.
- ❑ Si l'<expression> fournit la valeur zéro, l'exécution continue avec l'instruction qui suit le bloc d'instructions.
- ❑ Le <bloc d'instructions> est exécuté zéro ou plusieurs fois.
- ❑ La partie <expression> peut désigner :
 - ✓ Une variable de type numérique,
 - ✓ Une expression fournissant un résultat numérique.

Les structures répétitives

- ❑ La partie <bloc d'instructions> peut désigner :
 - ✓ Un bloc d'instructions compris entre accolades,
 - ✓ Une seule instruction terminée par un point virgule.

Exemple 1 :

```
/* Afficher les nombres de 0 à 9*/  
int i = 0 ;  
while (i<10)  
{  
    printf("%d\t",i) ;  
    i++ ;  
}
```

Exemple 2 :

```
/* Afficher les nombres de 10 à 1*/  
int i = 10 ;  
while (i)  
    printf("%d\t",i--);
```


Les structures répétitives

Exemple 3 :

Calculer et afficher la somme des diviseurs d'un entiers N sauf lui-même.

```
i=2;  
S=1;  
while (i<=N/2)  
{  
    if(N%i==0)  
        s+=i;  
    i++ ;  
}
```

Les structures répétitives

4. La structure « do while »

Syntaxe :

```
do  
{  
    < Bloc d'instructions> ;  
} while (<expression>) ;
```

- ❑ La boucle « do-while » teste sa condition après exécution de(s) instruction(s) du corps de la boucle.
- ❑ Le(s) instruction(s) du "**do... while**" sont effectuées au moins une fois et aussi longtemps que l'<expression> fournit une valeur différente de zéro.
- ❑ En pratique, la structure "do...while" n'est pas si fréquente que while ; mais dans certains cas, elle fournit une solution plus élégante.
 - Une application typique de do - while est la saisie de données qui doivent remplir une certaine condition.

Les structures répétitives

Exemple 1 : Saisir un entier entre 1 et 10

```
int N ;  
do  
{  
    printf(" Donner un nombre entre 1 et 10 :") ;  
    scanf("%d", &N) ;  
}while(N<1 || N>10) ;
```

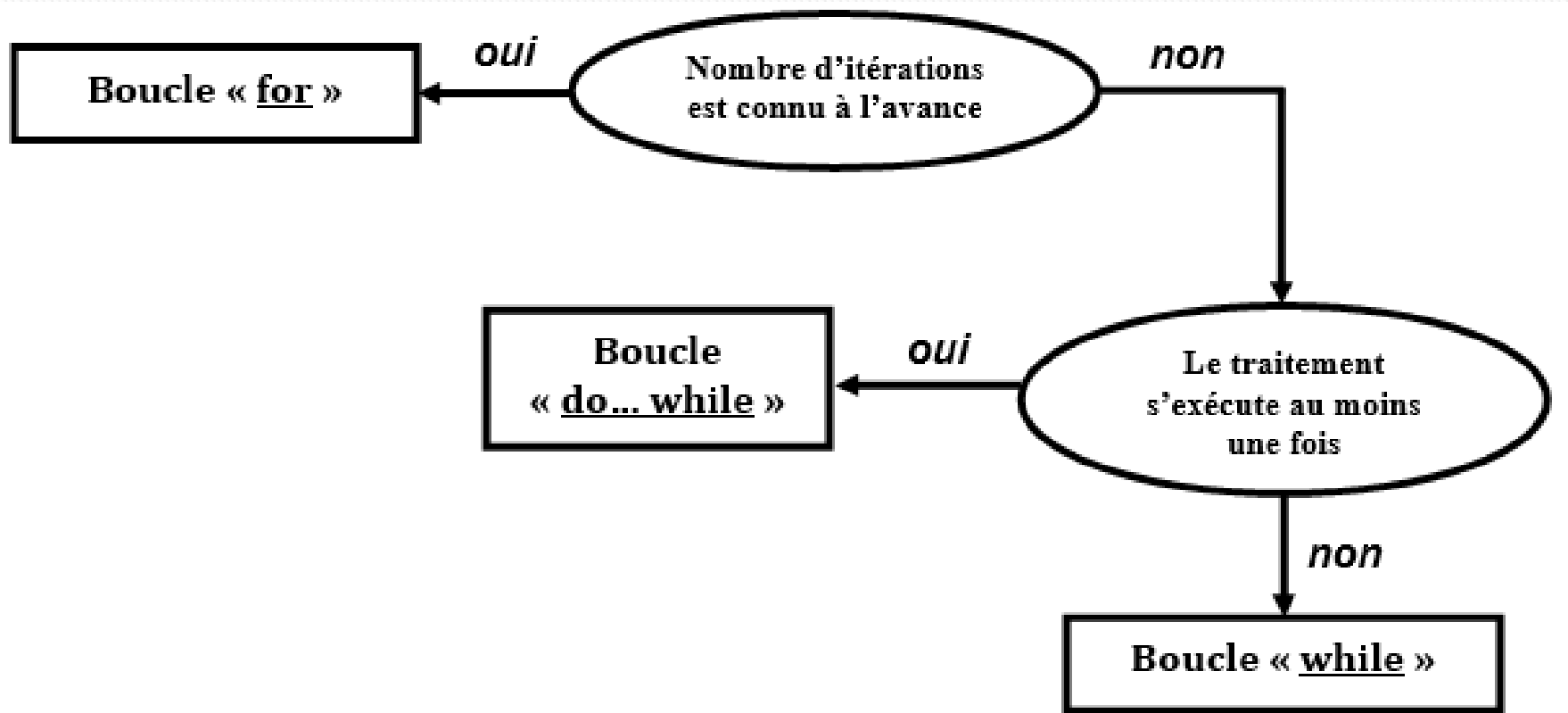
Les structures répétitives

Exemple 1 : Calculer et afficher la somme des diviseurs d'un entiers N(avec $N > 1$) sauf lui-même

```
void main()
{
    int i=2, S=1, N;
    do
    {
        printf(" Donner un entier >1: ");
        scanf("%d",&N) ;
    } while(n<=1);
    do
    {
        if(N%i==0)
            s+=i;
        i++ ;
    } while (i<=N/2);
    printf(" La somme = %d\n« , s);
}
```

Les structures répétitives

5. Choix de la structure répétitive



Les structures répétitives

6. Les instructions de branchement inconditionnel : "break", "continue" et "goto"

6.1 L'instruction "break"

- ☐ Le langage C autorise également l'emploi de cette instruction dans une boucle.
- ☐ Elle sert à interrompre le déroulement de la boucle, en passant à l'instruction qui suit cette boucle.

Les structures répétitives

Exemple :

```
#include<stdio.h>
void main()
{
    int i ;
    for ( i=1 ; i<=10 ; i++ )
    {
        printf ("début tour %d\n", i) ;
        printf ("bonjour\n") ;
        if ( i==3 )
            break ;
        printf ("fin tour %d\n", i) ;
    }
    printf ("après la boucle") ;
}
```

Exécution :

début tour 1
bonjour
fin tour 1
début tour 2
bonjour
fin tour 2
début tour 3
bonjour
après la boucle

Remarque : En cas de boucles "imbriquées", "break" fait sortir de la boucle la plus interne.

Les structures répétitives

6.2 L'instruction continue

L'instruction continue, quant à elle, permet de passer "prématurément" au tour de boucle suivant. En voici un premier exemple avec for :

Exemple :

```
#include<stdio.h>
void main()
{  int n ;
   do
   {   printf ("donnez un nb>0 : ") ;
       scanf ("%d", &n) ;
       if (n<0)
       {
           printf ("svp >0\n") ;
           continue ;
       }
       printf ("son carré est : %d\n", n*n) ;
   }while(n) ;
}
```

Exécution :

donnez un nb>0 : 4
son carré est : 16
donnez un nb>0 : -5
svp >0
donnez un nb>0 : 2
son carré est : 4
donnez un nb>0 : 0
son carré est : 0

Les structures répétitives

6.3 L'instruction "goto"

Elle permet "classiquement" le "branchement" en un emplacement quelconque du programme.

Exemple :

```
#include<stdio.h>
void main()
{   int i;
    for ( i=1 ; i<=10 ; i++ )
    {
        printf ("début tour %d\n", i) ;
        printf ("bonjour\n") ;
        if ( i==3 )
            goto sortie ;
        printf ("fin tour %d\n", i) ;
    }
    sortie : printf ("après la boucle") ;
}
```

Exécution :

début tour 1
bonjour
fin tour 1
début tour 2
bonjour
fin tour 2
début tour 3
bonjour
après la boucle