

## Travaux Pratiques N°4

### Les structures répétitives

#### Objectifs :

- Rappeler les structures itératives
- Présenter les syntaxes offertes par le langage C permettant la manipulation des structures itératives en C.

#### **I. La structure for .....**

##### Syntaxe :

```
for(initialisation ; condition de continuité vraie ; modification)
{
    ..... ;
    .....; /*bloc d'instructions*/
    .....;
}
```

##### Remarque :

- Les {} ne sont pas nécessaires lorsque le bloc ne comporte qu'une seule instruction.
- Les 3 instructions du for ne portent pas forcément sur la même variable.
- Une instruction peut être omise, mais pas les ;

##### Exemple :

```
p=0 ;
for (i=0; p<30; i++)
{
    ..... ;
    .....; /*bloc d'instructions*/
    ..... ;
    p=p + 2*i;
}
```

#### **II. La Structure while**

##### Syntaxe :

```
while (expression)
{
    ..... ;
    .....; /*bloc d'instructions*/
    .....;
}
```

Le test se fait d'abord, le bloc d'instructions n'est pas forcément exécuté.

##### Remarques :

- Les {} ne sont pas nécessaires lorsque le bloc ne comporte qu'une seule instruction.
- Le traitement s'exécute 0 ou n fois ! (suivant la condition)

### III. la Structure do ... while

#### Syntaxe :

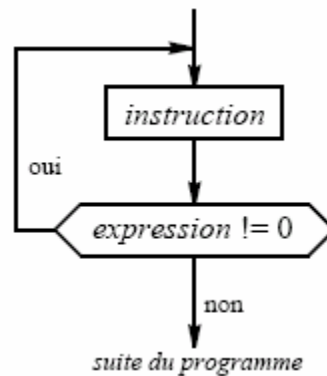
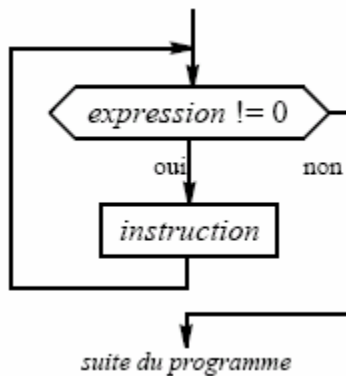
```
do{
    ..... ;
    .....; /*bloc d'instructions*/
    .....;
} while (expression) ;
```

Le test se faisant après, le bloc est exécuté au moins une fois

#### Remarques :

- les {} ne sont pas nécessaires lorsque le bloc ne comporte qu'une seule instruction.
- Le traitement s'exécute 1 ou n fois ! (suivant la condition)

Le fonctionnement de **while** et **do...while** est décrit par les organigrammes suivants :



#### Exemple:

```
int somme=0, i=1 ;
while (i<=5)
{
    somme += i ;
    i++ ;
}
```

```
int somme=0, i=1 ;
do
{
    somme += i ;
    i++ ;
} while (i<=5)
```

#### IV. Travail demandé

##### Exercice 1

Ecrire un programme qui permet de déterminer le minimum et le maximum de n nombres saisies au clavier.

##### Exercice 2:

Un entier naturel de trois chiffres est dit cubique s'il est égal à la somme des cubes de ses trois chiffres.

##### **Exemple :**

153 est cubique car  $153 = 1^3 + 5^3 + 3^3$

Réaliser en C un algorithme qui cherche et affiche tous les entiers cubiques de trois chiffres.

##### Exercice 3:

Écrire un programme C qui permet de lire un entier positif et déterminer tous ses facteurs premiers.

##### **Exemples:**

$30 = 2 * 3 * 5$

$36 = 2 * 2 * 3 * 3$

$99 = 3 * 3 * 11$

##### Exercice 4:

Deux nombres entiers sont premiers entre eux s'ils n'ont pas d'autres diviseurs communs que 1.

- 7 et 13 n'ont que 1 comme diviseur commun donc 7 et 13 sont premiers entre eux.
- 12 et 32 ont plusieurs diviseurs communs : 1 ; 2 et 4 donc 12 et 32 ne sont pas premiers entre eux.

Écrire un programme C qui saisit deux entiers N1 et N2, vérifie et affiche s'ils sont premiers entre eux ou non.

##### Exercice 5:

Un nombre est dit palindrome s'il est écrit de la même manière de gauche à droite ou de droite à gauche.

**Exemples :** 101 ; 22 ; 3663 ; 10801, etc.

Écrire un programme C permettant de déterminer et d'afficher tous les nombres palindromes compris dans l'intervalle [100..9999].

##### Exercice 6 :

Deux entiers N1 et N2 sont dits frères si chaque chiffre de N1 apparaît au moins une fois dans N2 et inversement.

##### **Exemples :**

- Si N1 = 1164 et N2 = 614 alors le programme affichera : N1 et N2 sont frères
- Si N1 = 405 et N2 = 554 alors le programme affichera : N1 et N2 ne sont pas frères

Écrire un programme C qui saisit deux entiers N1 et N2, vérifie et affiche s'ils sont frères ou non.