

Séance 6 : Instructions while et do...while

Après avoir vu les instructions conditionnelles avec « if », « else » et « switch », nous allons maintenant étudier les instructions de boucle. Ces instructions vont notamment nous permettre de programmer des algorithmes demandant des actions répétitives.

Instruction while

L'instruction while s'utilise comme suit :

```
while( expression)
{
    instructions...
}
```

L'expression est une expression entière que le compilateur évaluera comme une valeur logique (comme dans le cas de l'instruction « if »). L'ordinateur dans un premier temps évalue la valeur de l'expression, si cette expression est vraie, les instructions suivant immédiatement le « while » sont exécutées, puis l'expression est à nouveau évaluée et ainsi de suite tant que l'expression est vraie. Dès que l'expression est fausse, l'exécution du programme reprend immédiatement après le bloc constituant la boucle.

Voici un exemple d'utilisation :

```
i = 1;
while( i <= 10)
{
    instructions...
    i++;
}
```

Dans cet exemple, le bloc d'instructions est exécuté pour des valeurs de i allant de 1 à 10.

Exercice 1

Ecrire un programme affichant les carrés des entiers de 1 à 100.

Exercice 2

Ecrire un programme utilisant l'instruction « while » qui calcule la factorielle d'un nombre saisi par l'utilisateur. On utilisera d'abord le type « int » puis le type « long » afin de comparer leurs capacités respectives.

Instruction do...while

Le fonctionnement de l'instruction « do...while » est très similaire à celui de l'instruction « while ». On l'utilise comme suit :

```
do
{
    instructions...
} while( expression);
```

Contrairement au cas du « while », le bloc d'instructions est ici toujours au moins exécuté une fois. Puis seulement l'expression est évaluée, si elle est vraie on recommence, si elle est fausse, l'ordinateur continue l'exécution du programme. On notera la présence du point-virgule immédiatement après le while. Il est obligatoire.

Exercice 3

Réécrire l'exercice 2 en utilisant l'instruction « do...while ».

Exercice 4

Ecrire un programme qui demande à l'utilisateur d'entrer un nombre entre 1 et 100. Si l'utilisateur se trompe lui redemander de saisir un nombre et ce tant que le nombre n'est pas correct.

Exercice 5

Ecrire un programme `capital.c` qui demande à l'utilisateur de rentrer la valeur d'un capital initial qu'il souhaite placer, ainsi que le taux annuel auquel sera effectué le placement, et qui affiche la valeur de ce capital à la fin de chaque année, jusqu'à qu'il ait atteint deux fois la valeur du capital initial.