Les tableaux

Dans de très nombreux programmes, on a besoin d'avoir plusieurs variables du même type et qui jouent quasiment le même rôle. Pensez par exemple à la liste des utilisateurs d'un site web : cela représente une grande quantité de variables de type string. Ou les dix meilleurs scores de votre jeu, que vous stockerez dans différentes cases mémoires, toutes de type int.
Le C++, comme presque tous les langages de programmation, propose un moyen simple de regrouper des données identiques dans un seul paquet. Et comme l'indique le titre du chapitre, on appelle ces regroupements de variables des tableaux.

La déclaration d'un tableau est très similaire à celle d'une variable (figure suivante).

TYPE NOM [TAILLE] ;

On indique le type, puis le nom choisi et enfin, entre crochets, la taille du tableau. Voyons cela avec un exemple.

```
int main()
{
  int meilleurScore[5];       //Déclare un tableau de 5 int
  double anglesTriangle[3];   //Déclare un tableau de 3 double
  return 0;
}int main()
```

Accéder aux éléments d'un tableau

Chaque case d'un tableau peut être utilisée comme n'importe quelle autre variable, il n'y a aucune différence. Il faut juste y accéder d'une manière un peu spéciale. On doit indiquer le nom du tableau et le numéro de la case. Dans le tableau `meilleurScore`, on a accès à cinq variables : la première case de `meilleurScore`, la deuxième, etc, jusqu'à la cinquième.

Pour accéder à une case, on utilise la syntaxe `nomDuTableau[numeroDeLaCase]`. Il y a simplement une petite subtilité : la première case possède le numéro 0 et pas 1. Tout est en quelque sorte décalé de 1. Pour accéder à la troisième case de `meilleurScore` et y stocker une valeur, il faudra donc écrire :

```
meilleurScore[2] = 5;
```

Parcourir un tableau

Le gros point fort des tableaux, c'est qu'on peut les parcourir en utilisant une boucle. On peut ainsi effectuer une action sur chacune des cases d'un tableau, l'une après l'autre : par exemple afficher le contenu des cases.

On connaît *a priori* le nombre de cases du tableau, on peut donc utiliser une boucle `for`. Nous allons pouvoir utiliser la variable `i` de la boucle pour accéder au ième élément du tableau. C'est fou, on dirait que c'est fait pour !

```
int const nombreMeilleursScores(5);          //La taille du tableau
int meilleursScores[nombreMeilleursScores];   //Déclaration du tableau
meilleursScores[0] = 118218;  //Remplissage de la première case
meilleursScores[1] = 100432;  //Remplissage de la deuxième case
meilleursScores[2] = 87347;   //Remplissage de la troisième case
meilleursScores[3] = 64523;   //Remplissage de la quatrième case
meilleursScores[4] = 31415;   //Remplissage de la cinquième case
for(int i(0); i<nombreMeilleursScores; ++i)
{
   cout << meilleursScores[i] << endl;
}
```