**Chapitre 9 : Programmation – les tableaux spé ISN – TS**

En programmation, un tableau est un ensemble de cases mémoires dans lesquelles on peut stocker des variables, toutes de même type.

## I. Tableau à une dimension

Un tableau à une dimension (ligne ou colonne), comme toutes les variables, doit être déclaré avant d'être rempli et utilisé. **Sa taille est fixée dès le départ et ne doit pas changer par la suite.**

```
type_de_variable  nom_du_tableau[dimension_du_tableau] ;
```

Exemple : déclaration d'un tableau d'entiers de taille 10 :

```
int  tab[10] ;
```

Il est possible d'initialiser les valeurs du tableau au moment de la déclaration :

```
int  tab[10] ={1, 4, 72, 21, 0, 6, 19, 3, 15, 37} ;
int  tab2[10] = {0} ; // Toutes les cases du tableau sont initialisées à la valeur 0.
```

Pour parcourir le tableau, on utilise ensuite une boucle for.
À noter que les cases sont numérotées de 0 à dimension-1.

**Question 1 :** Indiquer le résultat de l'exécution du code suivant :

```
int  tab[10] = {1, 4, 72, 21, 0, 6, 19, 3, 15, 37} ;
int  i ;
for (i = 0 ; i < 10 ; i = i + 1) {
   printf("%d ", tab[i]) ;
}
```

Une boucle for similaire peut également permettre de remplir le tableau (ici, avec des 1) :

```
int  tab[10] ;
int i ;
for (i = 0 ; i < 10 ; i = i + 1) {
   tab[i] = 1 ;
}
```

**Question 2 :** Indiquer le contenu du tableau défini ci-dessous :

```
int  tab[10] ;
int i ;
for (i = 0 ; i < 10 ; i = i + 1) {
   tab[i] = i * i ;
}
```

## II. Tableau à deux dimensions

La déclaration se fait selon le mode suivant :

```
int  tab[10][15] ;
```

Pour parcourir le tableau, il faut maintenant deux boucles for imbriquées, une pour parcourir les lignes, et une autre pour parcourir l'ensemble des colonnes à chaque parcours de ligne.
On ne s'intéresse qu'au contenu des cases, et pas à la façon dont on les délimite.

**Question 3 :** Indiquer le contenu du tableau défini ci-dessous.

```
int  tab[2][3] ;
int  i, j ;
for (i = 0 ; i < 2 ; i = i + 1) {
   for (j = 0 ; j < 3 ; j = j + 1) {
      tab[i][j] = i + j ;
   }
}
```

**Question 4 :** Que fait le code suivant ? Quel est le résultat de son exécution ?

```
int  tab[4][3] = {{3, 5, 6}, {7, 9, 5}, {1, 8, 5}, {5, 6, 2}} ;
int  i, j, compteur = 0 ;
for (i = 0 ; i < 4 ; i = i + 1) {
   for (j = 0 ; j < 3 ; j = j + 1) {
      if (tab[i][j] == 5) {
        compteur = compteur + 1 ;
      }
   }
}
printf("%d", compteur) ;
```

**Question 5 :** Recopier le code suivant dans Code::Blocks et l'exécuter.

```
#include <stdio.h>
int main () {
   int tab[4][3] = {{3, 5, 6}, {7, 9, 5}, {1, 8, 5}, {5, 6, 2}} ;
   int i, j;
   for (i = 0; i < 4; i++) {
      for (j = 0; j < 3; j++) {
        printf("%d ", tab[i][j]);  // Ligne 1
      }
      printf("\n");                // Ligne 2
   }
   return 0 ;
}
```

À quoi sert l'espace dans le "%d " de la ligne 1 ? À quoi sert la ligne 2 ?

**Évaluation : répondre au questionnaire « Les tableaux » sur Socrative (ISNJVERNE).**