Travaux Pratiques N°5
Les Tableaux à une dimension

Objectifs

- Définition et utilisation classique des tableaux.
- Savoir coder en langage C des répétitions pour les tableaux
- Modifier l'ordre des éléments d'un tableau (ajout, suppression, affichage des éléments).

I. Déclaration

Syntaxe :

<type simple> <Nom tableau> [<dimension>] ;

Exemples *:*

```
int T [20] ;
float T1 [100] ;
char T2 [30] ;
```

II. Mémorisation

En C, le nom d'un tableau est le représentant de l'adresse du premier élément du tableau. Les adresses des autres composantes sont calculées (automatiquement) relativement à cette adresse.

Exemple : int T [5] = {100,200,300,400,500} ;

……	**100**	**200**	**300**	**400**	**500**	……
Adresse :	1E06	1E08	1E0A	1E0C	1E0E	1E10

T

Réservation automatique

Si la dimension n'est pas indiquée explicitement lors de l'initialisation, alors l'ordinateur réserve automatiquement le nombre d'octets nécessaires.

Exemples

int A [ ] = {10, 20, 30, 40, 50};
==> réservation de **5*sizeof(int)** octets (dans notre cas: 10 octets)
float B [ ] = {-1.05, 3.33, 87e-5, -12.3E4};
==> réservation de **4*sizeof(float)** octets (dans notre cas: 16 octets)
char C [ ] = {'a', 'b', 'c', 'd', 'e'};
==> réservation de **5*sizeof(char)** octets (dans notre cas: 5 octets)

III. Accès aux composantes d'un tableau

Considérons un tableau T de dimension **N**:

Algorithmiquement,

- l'accès au premier élément du tableau se fait par **T[1]**
- l'accès au dernier élément du tableau se fait par **T[N]**

En C,

- l'accès au premier élément du tableau se fait par **T[0]**
- l'accès au dernier élément du tableau se fait par **T[N-1]**

Exemple : int T [5] = {100,200,300,400,500} ;

Nom : T	**100**	**200**	**300**	**400**	**500**
Indice :	**0**	**1**	**2**	**3**	**4**
Contenu	**T[0]**	**T[1]**	**T[2]**	**T[3]**	**T[4]**

IV. Chargement et affichage d'un tableau

```
#include <stdio.h>
void main(){
        int T [50]; /* déclaration d'un tableau de 50 cases */
        int N ; /* le nombre effectif d'éléments  */
        int i;  /* un compteur */
        do{
                printf ("donner le nombre d'éléments \n") ;
                scanf("%d", &N);
        } while (N<= 0 | | N>50) ;

        for (i=0; i<N; i++) { /* chargement du tableau */
                printf("T[%d]:",i);
                scanf("%d", &T[i]);
         }

        for (i=0; i<N; i++)  /* affichage des elements du tableaux */
        printf("T[%d] = %d \t ", i, T[i]);
}
```

VI. Travail demandé

Exercice 1
Ecrire un programme qui lit la dimension N d'un tableau T du type int (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau. Calculer et afficher ensuite la somme des éléments du tableau.

Exercice 2
Ecrire un programme qui lit la dimension N d'un tableau T du type int (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau. Effacer ensuite toutes les occurrences de la valeur 0 dans le tableau T et tasser les éléments restants. Afficher le tableau résultant.

Exercice 3
Ecrire un programme qui lit la dimension N d'un tableau T du type int (dimension maximale: 50 composantes), remplit le tableau par des valeurs entrées au clavier et affiche le tableau. Ranger ensuite les éléments du tableau T dans l'ordre inverse sans utiliser de tableau d'aide. Afficher le tableau résultant.
Idée: Echanger les éléments du tableau à l'aide de deux indices qui parcourent le tableau en commençant respectivement au début et à la fin du tableau et qui se rencontrent en son milieu.