

CHAPITRE 6

LES STRUCTURES EN C

I.DÉFINITION D'UNE STRUCTURE

Une structure étant un regroupement d'objets, la première chose à réaliser est la description de celle-ci (techniquement, sa **définition**), c'est-à-dire préciser de quel(s) objet(s) cette dernière va se composer. POUR DÉFINIR UNE STRUCTURE EN ON UTILISE **struct**

La syntaxe de toute définition est la suivante.

struct nom_type

{

Déclaration des variables qui composent le type

}

Exemple

Première méthode

struct personne

{

int matricule ;

char nom[20] ;

char prenom[50] ;

int age ;

char sexe;

};

personne est un nouveau type composé de deux variables de type int deux variables de type chaîne de caractères et une variable de type char.

struct personne p1,p2;qui déclare deux variables de type struct personne de noms p1 et p2;

Deuxième méthode

On peut déclarer des variables de type structure sans utiliser d'étiquette de structure, par exemple :

struct

{

int matricule ;

char nom[20] ;

char prenom[50] ;

int age ;

char sexe;

}p1,p2;

déclare deux variables de noms p1 et p2 comme étant deux structures de cinq membres, mais elle ne donne pas de nom au type de la structure. L'inconvénient de cette méthode est qu'il sera par la suite impossible de déclarer une autre variable du même type. En effet, si plus loin on écrit :

struct

{

int matricule ;

char nom[20] ;

char prenom[50] ;

int age ;

char sexe;

}p3;

Troisième méthode

On peut combiner déclaration d'étiquette de structure et déclaration de variables, comme ceci :

struct personne

```
{  
    int matricule ;  
    char nom[20] ;  
    char prenom[50] ;  
    int age ;  
    char sexe;  
}p1,p2;
```

déclare les deux variables p1 et p2 et donne le nom personne à la structure. Là aussi, on pourra utiliser ultérieurement le nom struct personne pour déclarer d'autres variables :

struct personne pers1, pers2, pers3;

qui seront du même type que p1 et p2.

Une structure peut être initialisée par une liste d'expressions constantes à la manière des initialisations de tableau. Exemple :

struct personne p = {350, "NDIAYE", "Fatoumata", 20, 'F'};

II. ACCES AUX MEMBRES DES STRUCTURES

Pour désigner un membre d'une structure, il faut utiliser l'opérateur de sélection de membre qui se note **'.'** (**point**). Par exemple, si p1 et p2 sont deux variables de type struct personne, on désignera le membre nom de p1 par **p1.nom** et on désignera le membre matricule de p2 par **p2.matricule**. **Les membres ainsi désignés se comportent comme n'importe quelle variable** et par exemple, pour accéder au premier caractère du nom de p2, on écrira : **p2.nom[0]**.

```
p1.nom= "thiandoum";  
scanf("%d",&p1.matricule);  
gets(p1.prenom);  
printf("%d",p1.matricule) ;  
puts(p1.nom) ;  
scanf("%c",p1.sexe);
```

III. TABLEAU DE STRUCTURE

Nous avons déjà utilisé les tableaux, cependant, leurs éléments étaient alors d'un type scalaire (int float ou char). En C il est possible de définir des tableaux dont les éléments sont des d'un type structure.

Déclaration :

struct personne etudiant[5] ;

Nous avons un tableau etudiant de 5 personnes

| | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| ❖ Matricul e | ❖ Matricul e | ❖ Matricul e | ❖ Matricul e | ❖ Matricul e |
| ❖ Nom | ❖ Nom | ❖ Nom | ❖ Nom | ❖ Nom |
| ❖ Prenom | ❖ Prenom | ❖ Prenom | ❖ Prenom | ❖ Prenom |
| ❖ Age | ❖ Age | ❖ Age | ❖ Age | ❖ Age |
| ❖ sexe | ❖ sexe | ❖ sexe | ❖ sexe | ❖ sexe |

Etudiant[0].matricule=1230 signifie que le matricule du premier est 1230

Etudiant[4].prenom="Adama" signifie le dernier se nomme Adama

Etudiant[2].sexe='F' signifie le troisième est de sexe féminin