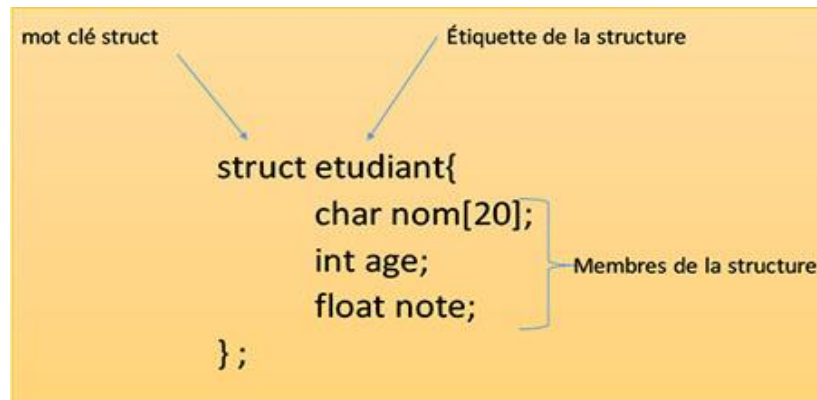


# TP 10 Structures

## À retenir

Une structure est un type de données défini par l'utilisateur en C / C++. Une structure crée un type de données qui peut être utilisé pour grouper des éléments de types éventuellement différents en un seul type.



## Comment créer une **structure**?

Le mot-clé '**struct**' est utilisé pour créer une structure.

```
1 struct personne{
2     char nom[20];
3     char adresse [60];
4     int age;
5     float salaire;
6 };
```

## Comment déclarer des variables de type **structure**?

Une variable de structure peut être déclarée avec une déclaration de structure ou comme une déclaration séparée comme les types de base.

```
1 // Une déclaration variable
2 // avec déclaration de structure.
3 struct point{
4     int x,y;
5 }p1;
6 // La variable p1 est déclaré avec "point"
```

```
1 // Une déclaration variable comme
2 // les types de données de base avec
3 // déclaration de structure.
4 struct point{
5     int x,y;
6 };
7
8 int main(void){
9     struct point p1;
10    // La variable p1 est déclarée
11    // comme une variable normale
12    return 0;
13 }
```

## Comment initialiser les membres de la **structure**?

Les membres de structure ne peuvent pas être initialisés dans la déclaration.

Les membres de structure peuvent être initialisés à l'aide d'accolades '{}'.  
C'est la seule manière d'initialiser les membres d'une structure.

```
1 struct point{
2     int x,y;
3 };
4
5 int main(void){
6     // Une initialisation valide. membre x obtient la valeur 0 et y
7     // obtient la valeur 1. L'ordre de déclaration est respecté.
8
9     struct point p1={0,1};
10    //
11    return 0;
12 }
```

## Comment accéder aux éléments de la **structure**?

Les membres de la Structure sont accessibles à l'aide de l'opérateur point (.)

```
1 #include <stdio.h>
2 struct point{
3     int x,y;
4 };
5
6 int main(void){
7     struct point p1={0,1};
8     p1.x=10;
9     printf ("x = %d, y = %d", p1.x, p1.y);
10    return 0;
11 }
```

## Qu'est-ce qu'un tableau de **structures** ?

Comme d'autres types de données primitifs, nous pouvons créer un tableau de structures.

```
1 #include< stdio.h>
2 struct Point{
3     int x, y;
4 };
5 int main(void){
6     // Créer un tableau de structures
7     struct Point tab[10];
8
9     // Accéder aux membres du tableau
10    // Premier point
11    tab[0].x=2;
12    tab[0].y=3;
13
14    // afficher les coordonnées du premier point
15    printf ("coordonnées du premier point; x = %d, y = %d", tab[0].x, tab[0].y);
16    return 0;
17 }
```

## Qu'est-ce qu'un pointeur de structures ?

Nous pouvons utiliser un pointeur sur une structure. Si nous avons un pointeur sur la structure, les membres sont accessibles à l'aide de l'opérateur (->).

```

1 #include< stdio.h>
2 struct Point{
3     int x, y;
4 };
5 int main(void){
6     struct Point p1={0,1};
7
8     // p2 est un pointeur sur la structure p1
9     struct Point *p2=&p1;
10
11     printf ("x = %d, y = %d", p2->x, p2->y);
12     return 0;
13 }
```

```

1 #include < stdio.h>
2
3 struct etudiant{
4     char prenom[20];
5     int age;
6 };
7 int main(void){
8
9     // pointeur et1 de type etudiant
10    struct etudiant *et1;
11    struct etudiant et2;
12
13    // initialisation
14    et1=&et2;
15
16    printf("Saisir votre prénom : ");
17    scanf("%s",&et1->prenom);
18
19    printf("saisir votre age : ");
20    scanf("%d",&et1->age);
21
22    printf("voici vos infos : ");
23    printf("Prénom : %s",et1->prenom);
24    printf("age : %d",et1->age);
25
26    return 0;
27 }
```

l'adresse de et2 est stockée dans le pointeur et1 à l'aide de **et1 = &et2** ;  
vous pouvez accéder aux membres de **et2** en utilisant le pointeur **et1**.

- **et1->prenom** équivalente à **(\*et1).prenom**
- **et1->age** équivalente à **(\*et1).age**

## Exercice 1

Ecrire un programme C qui définit une structure *point* qui contiendra les deux coordonnées d'un point du plan. Puis lit deux points et affiche la distance entre ces deux derniers.

La distance entre deux points A et B  
est donnée par :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

## Exercice 2

Ecrire un programme C qui définit une structure *etudiant* où un étudiant est représenté par son nom, son prénom et une note. Lit ensuite une liste d'étudiants entrée par l'utilisateur et affiche les noms de tous les étudiants ayant une note supérieure ou égale à 10 sur 20.

## Exercice 3

Ecrire un programme C, qui lit le nom, le prénom et l'âge de plusieurs personnes dans un tableau de structures, puis insère une nouvelle personne dans une position entrée par l'utilisateur.

## Exercice 4

Ecrire un programme C qui lit un ensemble de personnes avec leurs âges, dans un tableau de structures, et supprime ensuite toutes celles qui sont âgées de vingt ans et plus.

## Exercice 5

A l'aide d'un tableau de personnes caractérisée chacun par :

- Nom
- Prenom
- Numero de la rue
- Ville
- Numéro de téléphone
- La préfecture

Faire un programme permettant de :

- Saisir les contacts
- Rechercher automatiquement toutes les informations sur les personnes répondant à une valeur d'une rubrique donnée.

Lexique :

T : Tableau d'enregistrement

I : Indice de T

N : Nombre de contact

Nom : Chaîne de caractère

Prenom : Chaîne de caractère

Num : Numéro de la rue

Pref : préfecture

Ville : La ville

Tel : Le numéro de téléphone

Xville : La ville à chercher

Trouve : La variable booléenne

## Exercice 6

Définir la structure stagiaire constitué des champs suivants :

Champ	Type	
Nom	Chaine	
Prénom	Chaine	
DateNaissance	Structure	<div> <div></div> <div></div> <div></div> </div> <div> <i>jour</i>  <i>mois</i>  <i>annee</i> </div>

Le champ dateNaissance est aussi une structure dont les champs sont :

Champ	Type	
Jour	Entier	
Mois	Entier	
Année	Entier	

*nom*  
*prenom*  
*naissance*

*jour*  
*mois*  
*annee*

Ecrivez ensuite l'algorithme qui permet de lire et après afficher le nom, prénom et date de naissance d'un stagiaire

## Exercice 7

Soit un stock de 1000 articles dont chacun est caractérisé par sa référence, son nom et son prix.

- 1- Saisir les données du stock
- 2- Trouver toutes les références des articles dont le prix est compris entre deux prix donnés.
- 3- Trouver le prix d'un article dont la référence est données.

## Exercice 8

### Partie 1 :

- Définir une structure de type date comportant 3 entiers appelés jour, mois, annee.
- Définir une structure de type compte comportant :
  - 3 chaînes de 80 caractères      appelés nom, rue, ville
  - 1 entier      appelé no\_compte
  - 1 caractère      appelé etat\_compte
  - 1 réel      appelé versement
  - 1 structure de type date      appelé der\_versement
- Déclarer un tableau client de 100 éléments de type compte.

### Partie 2 :

- Faire saisir par l'utilisateur le nombre de clients à traiter.
- Lire pour chaque client chacune des informations les concernant : nom jusqu'à der\_versement (saisi sous la forme jj/mm/aaaa).

### Partie 3 :

Afficher les informations concernant chaque client sous la forme :

```

Client n°      :
Nom           :
Rue           :
Ville         :
Compte n°     :
Etat du compte :
Versement     :
Date de versement :
```