



Langage de programmation C :

Les Structures

-Les structures

- Une structure est une suite finie d'objets de types différents. Qui sont regroupé au sein d'une même entité.
- Contrairement aux tableaux, les différents éléments d'une structure n'occupent pas nécessairement des zones contiguës en mémoire.
- Chaque élément de la structure, appelé **membre** ou **champ**, est désigné par un identificateur.

```
struct enreg {  
    int numero;  
    int qte;  
    float prix;  
};
```

Les structures

- L'utilisation pratique d'une structure se déroule de la façon suivante :

1. On commence par déclarer la structure elle-même. Le modèle général de cette déclaration est le suivant :

```
struct nom_structure {  
    type_1 membre_1 ;  
    type_2 membre_2 ;  
  
    ...  
    type_n membre_n ;  
};
```

Les structures

2. Pour déclarer un objet de type structure correspondant au modèle précédent, on utilise la syntaxe :

```
struct nom_structure identificateur_objet ;
```

Ou bien, si le modèle n'a pas encore été déclaré au préalable :

```
struct nom_structure {  
    type_1 membre_1 ;  
    type_2 membre_2 ;  
    ...  
    type_n membre_n ;  
} identificateur_objet ;
```

3. On accède aux différents membres d'une structure grâce à l'opérateur membre de structure, noté ".". Le i-ème membre de objet est désigné par l'expression :

```
identificateur_objet.membre_i
```

Les types dérivés

- **struct** **enreg** **art1**,**art2**; */* déclaration */*
- **art1**.numero=15; */*affectation*/*
- printf("%f",**art1**.prix); */* affiche la valeur du champ prix*/*
- scanf("%f",&**art2**.prix); */*lit une valeur qui sera affecté
au champ prix de la structure art2 */*
- **art1**.numero++; */* incrémente de 1 la valeur du champ
numero de la structure art1*/*

Les structures

- Il est possible d'affecter à une structure le contenu d'une structure définie à partir du **même modèle**. Par exemple nous pouvons écrire:
 - `art1=art2;`
- Une telle affectation remplace :
 - `art1.numero = art2.numero;`
 - `art1.qte = art2.qte;`
 - `art1.prix = art2.prix;`
- Initialisation d'une structure au moment de sa déclaration :
 - `struct` enreg `art1`={100,290,3000};

Les structures

- structure comportant des tableaux:

- `struct` personne{ `char` nom[30];
`char` prenom[20];
`float` heures[31];
} employe, courant;

*/*heures: nbre d'heures du travail dans chaque jours du mois*/*

- `employe.heures[4];` */*5^{ème} éléments du tableau heures*/*
 - `employe.nom[0];` */* premier caractère du champ nom*/*
- Initialisation:

`struct` personne courant={"Dupont", "julea"}

Les structures

- Tableaux de structures

```
struct point { char nom;  
               int x;  
               int y;  
            };
```

```
struct point courbe[50];
```

- Cette structure sert à représenter un point d'un plan qui serait défini par son nom(caractère) et ses deux coordonnées.
- `courbe[i].nom;` `courbe[i].x;` `courbe[i].y;` */*représente le nom du point de rang i, la valeur du champ x et la valeur du champ y*/*
- Initialisation :
 - `struct point courbe[50]={{'A',10,25},{ 'M',13,34},{ 'R',2,6}};`

Les structures

- Structures comportant d'autres structures:
 - Supposant que l'intérieur de la structure `employe` et `courant` nous ayant besoin d'introduire deux dates: la date d'embauche et la date d'entrée. Si ces dates sont elles-mêmes des structures comportant trois champs correspondant au jour et à l'année, nous pouvons alors procéder aux déclarations suivantes :
 - `struct date` { int jour;
 int mois;
 int annee};
 - `struct personne` { char nom[30];
 char prenom[20];
 float heures[31];
 struct date date_embauche;
 struct date date_poste;
 } employe, courant;

Les structures

- La notation :
 - `employe.date_embauche.annee`

Représente l'année d'embauche correspondant à la structure `employe`. Il s'agit d'une valeur de type `int`.

- `courant.date_embauche`

Représente la date d'embauche de la structure `courant`, et nous pouvons faire l'affectation:

- `courant.date_embauche=employe.date_poste;`

Les structures

- Exercice:

- Ecrire un programme qui lit au clavier des informations dans un tableau de structure du type point défini comme suit:

```
struct point{  
    int num;  
    float x;  
    float y;  
}
```

Le nombre d'éléments du tableau sera fixé par une instruction #define.

Affiche à l'écran l'ensemble des informations précédentes?