

Les structures de données

1. Rôle des structures de données

Dans un tableau, tous les constituants doivent obligatoirement être du même type, Alors qu'une structure rassemble des variables, qui peuvent être de types différents, sous un seul nom ce qui permet de les manipuler facilement.

Exemples :

- ⊗ Données des élèves (CNE, Nom, Prénom, Notes, ...)
- ⊗ La date est composée d'un jour, d'un mois et d'une année.

2. Définition d'une structure de données

Définir une structure consiste en fait à définir un nouveau type.

```
struct complexe {
    float reel;
    float imag;
};

struct complexe C1, C2;
```

Dans cet exemple, on définit une structure contenant deux réels, puis on déclare deux variables ayant pour type **struct complexe**.

Remarque : Le mot clé *typedef* pour simplifier le nom du type *struct complexe* ;

```
typedef struct complexe Comp ;
```

Dans ce cas on peut déclarer les variables de ce type comme suite : **Comp C1, C2 ;**

3. Manipulation des enregistrements

On accède à la valeur d'un membre d'une structure en faisant suivre l'identificateur de la variable de type structure par un point suivi du nom du membre auquel on souhaite accéder.

Exemple :

```
C1.reel = 2 ; /* permet d'accéder au membre 'reel' de la variable C1. */
C1.imag = 7 ; /* permet d'accéder au membre 'imag' de la variable C1. */
```

On peut initialiser une structure au moment de sa déclaration, par exemple :

```
struct complexe X={10,5};    ou bien    Comp X = {10, 5} ;
```

Remarque : $C2 = C1$; est équivalent à : $C2.reel = C1.reel$; $C2.imag = C1.imag$;

4. Accès aux structures de données à l'aide des pointeurs

La manipulation des structures de données est possible à l'aide des pointeurs.

Exemple :

```
struct complexe C, *P ;
P = &C ;
```

Dans ce cas l'accès aux membres de la structure se fait à l'aide de l'opérateur \rightarrow au lieu de point.

$P\rightarrow reel = 10$; est équivalent à $C.reel = 10$;

Exercice 1 :

Un étudiant est défini par la structure suivante :

```

struct Etudiant {
    int   cne ;
    char nom[30] ;
    char prenom[30] ;
    int   age ;
};

```

1. Ecrire un programme qui permet de créer une variable de type Etudiant, puis remplir (à partir du clavier) et afficher les informations de cet étudiant.
2. Ecrire un programme qui permet de saisir les informations d'une classe dans un tableau (de 20 Etudiants), puis il affiche l'âge moyen de la classe.
3. Refaire la question précédente on utilisant un pointeur pour parcourir les éléments du tableau des étudiants.

5. Les structures imbriquées

Les membres d'une structure peuvent être des structures, par exemple, dans la structure précédente, on peut remplacer le membre **age** par la structure **Date** suivante :

```

struct date {
    int jour ;
    int mois ;
    int annee ;
};

```

La structure Etudiant sera définie comme suite :

```

struct Etudiant {
    int   cne ;
    char  nom[30] ;
    char  prenom[30] ;
    struct date DateNais ;
};

```

Déclaration d'un étudiant :

```

struct Etudiant E, *P ;

```

Manipulation des membres :

```

    E.cne = 23456 ;
    E.DateNais.jour = 5 ;
    E.DateNais.mois = 11 ;

```

Ou bien

```

    P = &E ;
    P->cne = 23456 ;
    P->DateNais.jour = 5 ;
    P->DateNais.mois = 11 ;

```

Exercice 2 :

Refaire l'exercice 1 en utilisant la nouvelle structure imbriquée **Etudiant**.