Chapitre 6 : Valeurs composées

Informatique de base

2013-2014

Sup Galilée

Qu'est-ce qu'une valeur composée ?

Une valeur composée est une aggrégation de données qui peuvent être simples (des données numériques) ou composées, de même type ou de types différents. Par exemple, la liste des températures moyennes de chaque mois d'une année (tableau de douze flottants) ou bien une fiche individuelle composée d'un nom, d'une liste de prénoms et d'un âge (structure).

En C, on distingue :

- les structures qui sont composées de données de types différents,
- les tableaux qui sont composées de données de même type.

Structures

Une structure est une suite contiguë de données de types qui peuvent être différents et qui constituent les champs de cette structure.

Si ident est un identificateur et decl-1, ..., decl-n sont des déclarations de variables, alors la déclaration suivante

```
struct ident
 {
 decl-1;
 ...
 decl-n;
 };
```

est une structure.

Structures

Une déclaration de structure permet de définir un nouveau type appelé ident et composé de ses différents champs.
Par exemple,

```
struct point
 {
 char nom;
 float x,y;
 };
```

définit le type `point` composé de trois champs. On peut alors définir des variables `A` et `B` de type `point` de la façon suivante :
struct point A, B; On accède aux champs de ces variables par exemple comme ceci : `A.nom`, `B.x`, `A.y`.
On peut donc par exemple attribuer une valeur à la variable `A` comme ceci :

```
A.nom = 'a'; A.x = 0.5; A.y = 1.7;
```

Structures

Les structures peuvent être imbriquées comme par exemple :

```
struct date
  {
  int jour, mois, annee;
  };
struct vacances
  {
  char periode;
  char zone;
  struct date debut;
  struct date fin;
  };
```

Valeur et affectation d'une valeur

Si nom-variable-structure est le nom d'une variable dont le type est une structure, et si ident est le nom d'un champ de la structure, dont le type est T, alors `nom-variable-structure.ident` est une expression telle que

- type(`nom-variable-structure.ident`)=T,
- val(`nom-variable-structure.ident`) est la valeur du champ `ident` de la variable `nom-variable-structure`.

Supposons par exemple que l'on ait déclaré une variable `v` par `struct vacances v;`. On peut lui donner des valeurs en accédant à ses champs :

```
v.periode=3; v.zone='B'; v.debut.jour=1; v.debut.mois=7;
v.debut.annee=2014; v.fin.jour=31; v.fin.mois=8;
v.fin.annee=2014;
```

Tableaux

En C, un tableau est une suite contiguë de données de même type qui constituent les éléments du tableau.

Si T est un type, nom-tableau est un identificateur, n est un entier littéral, $\texttt{exp}_0$, ..., $\texttt{exp}_{n-1}$ sont des expressions constantes de type T, alors la déclaration
`T nom-tableau[n] = {` $\texttt{exp}_0$`,...,` $\texttt{exp}_{n-1}$`};`
définit un tableau de nom nom-tableau composé de n éléments de type T chacun. Ces éléments sont numérotés de 0 à $n-1$ (et non de 1 à n). Ils sont initialisés respectivement avec les valeurs val($\texttt{exp}_0$), ..., val($\texttt{exp}_{n-1}$).

L'initialisation est facultative. On peut donc par exemple effectuer la déclaration suivante `T nom-tableau[n];`

Remarque

En fait l'identificateur nom-tableau n'est pas le nom du tableau. C'est le nom d'une constante dont la valeur est l'adresse du premier élément (case numéro 0) de ce tableau (c'est un pointeur).

L'expression nom-tableau n'est donc pas une valeur gauche. C'est pour cela qu'on ne peut pas, en langage C, affecter un tableau à un autre à l'aide d'une expression de la forme `nom-tableau1=nom-tableau2;`

Si i est un entier entre 0 et n-1, alors nom-tableau[i] est une variable de type T.

Par exemple, si on définit un tableau de flottants `float Temp_moyenne[12];` alors on peut attribuer une valeur à chaque case du tableau comme par exemple

```
Temp_moyenne[0]=16.3;
...
Temp_moyenne[6]=26.1;
...
Temp_moyenne[11]=16.5;
```