LEÇON 3 : LES ENREGISTREMENTS

SANE ARNAUD

Table des matières

1- Structure d'enregistrement

Définition : 1.1- Définition

Un enregistrement, appelé structure En langage C, est une variable complexe qui permet de désigner sous un seul nom un ensemble de valeurs pouvant être de types différents.

1.2- Caractérisation

- Chaque élément de la structure est nommé champ
- L'accès à un champ se fait par son nom dans la structure

Avant de déclarer une variable structure , il faut au préalable créer son type ainsi que les différents champs qui le compose.

2- Syntaxe

Syntaxe : 2.1- Déclaration d'une structure

Lors de la déclaration de la structure, on définit un modèle où on indique les champs de la structure, c'est-à-dire le type et le nom des variables qui la composent :

Syntaxe:

struct <Nom_ Structure> {

<type_ champ1> <Nom_ Champ1>;

<type_ champ2> <Nom_ Champ2>;

<type_ champ3> <Nom_ Champ3>;

...

} :

La déclaration d'une structure est constituée de :

- Du mot réservé struct qui annonce l
- Du type et du nom a créé

Exemple

Énoncé :

Déclarer une structure représentant une personne

Résolution :

```
1
2 struct Personne{
3 char nom[20];
4 char prenom[50];
5 int age;
6 char cel[15];
7 };
8
```

Remarque :

- La déclaration d'une structure ne fait que donner l'allure de la structure, c'est-à-dire en quelque sorte une définition d'un type de variable complexe.
- La déclaration ne réserve donc pas d'espace mémoire pour une variable structurée (variable de type structure).
- Il faut donc définir une (ou plusieurs) variable(s) structurée(s) après avoir déclarée la structure

Syntaxe : 2.2- Déclaration d'une variable de type structure

La définition d'une variable structurée est une opération qui consiste à créer une variable ayant comme type celui d'une structure que l'on a précédemment déclaré, c'est-à-dire la nommer et lui réserver un emplacement en mémoire.

Une variable structurée doit être définie comme suit :

struct <Nom_ Structure> <Var_ Structure>;

Exemple

Énoncé :

Déclarer les variables p1 et p2 de type structure de personne

Résolution :

```
CAS 1 :
struct Personne{
char nom[20];
char prenom[50];
int age;
char cel[15];
};
struct Personne p1,p2;
CAS 2 :
struct Personne{
char nom[20];
char prenom[50];
int age;
char cel[15];
}p1,p2;

```

Application 2 :

Exercice 1

Sujet :

On veut créer un enregistrement personnel avec les champs suivant : nom,prenoms,sexe,salaire,nbreenfant. Remplissez les trous par les valeurs attendues.

NB :

1. La création des champs doit respecter l'ordre de leurs énumérations ci-dessus.
2. Toutes les réponses doivent être en minuscule

{

[20]

[50]

[1]

}

3- Manipulation d'une structure

IV

3.1 Accès aux champs d'une structure

Pour accéder aux champs d'une structure, il suffit d'appeler le nom de la variable suivie de l'opérateur point (.), ensuite on spécifie le nom du champ a manipulé.

<Var_ Structure>.<Nom_ Champi>;

Exemple

Nous allons utiliser la structure précédente dans notre exemple.

```
1 struct Personne{
2 char nom[20];
3 char prenom[50];
4 int age;
5 char cel[15];
6 };
7 struct Personne p1;
8  ///***************************** Pour accéder au différents champs
  ****************************
9
10 p1.nom; // accéder au nom
11 p1.prenom // accéder au prénom
12
```

3.2 Affectation

L'affectation de valeur aux différents champs d'un enregistrement se fait comme suit :

Nonvariable.champ = valeur ;

Exemple :

Nous utiliserons la variable déclarée ci-dessus.

```
1 P1.prenom = "Jean";
```

Exemple

Application :

Écrire un programme qui permet de donner l'aîné de deux personnes saisies au préalable par l'utilisateur.

Résolution :

```
1 #include <stdio.h>
2 struct Personne
3 {
```

```c
char nom[21];
char prenom[50];
int age;
char cel[15];
};
int main()
{
struct Personne p1,p2;
//************** Permet de rentrer les informations de la première personne **************
printf("SAISIE LES INFORMATIONS DE LA PREMIERE PERSONNE\n");
printf("NOM: ");gets(p1.nom);
printf("PRENOM: ");gets(p1.prenom);
printf("AGE: ");scanf("%d",&p1.age);
printf("CEL: ");gets(p1.cel);
//************** Permet de rentrer les informations de la deuxième personne **************
printf("SAISIE LES INFORMATIONS DE LA DEUXIEME PERSONNE\n");
printf("NOM: ");gets(p2.nom);
printf("PRENOM: ");gets(p2.prenom);
printf("AGE: ");scanf("%d",&p2.age);
printf("CEL: ");gets(p2.cel);
//************** Permet de faire la vérification pour afficher le plus âgé **************
if(p1.age > p2.age)
{printf("NOM: %s PRENOM: %s AGE: %2d CEL: %s",p1.nom,p1.prenom,p1.age,p1.cel);
}
else
{
 if (p1.age < p2.age)
 {printf("NOM: %s PRENOM: %s AGE: %2d CEL: %s",p2.nom,p2.prenom,p2.age,p2.cel);
}
 else
 {
  printf("Les deux ont le même âge ");
}
}

return 0;
}
```

Application 3 :

V

Exercice 1

```
#include <stdio.h>
struct personnel{
char nom[20];
char prenom[50];
char sexe[1];
int nbenf ;
int salaire;
};
int main()
{
struct personnel empl ;
printf("Entrer le nom de l'employé") ;
printf("Entrer le prénom de l'employé") ;
printf("Entrer le sexe de l'employé") ;
printf("Entrer le statut de l'employé") ;
//*** Faite une affectation de 950000 au salaire

return 0 ;
}
```

4 - Travaux dirigés

Énoncé :

Écrire un programme en C qui permet de saisir 10 étudiants à l'aide d'un enregistrement étudiant ayant pour champs nom, prénoms, matricule et moyenne. Le programme devra donner la liste des étudiants ayant une moyenne supérieure à 12.

Solution :

```c
#include <stdio.h>
struct Etudiant
{
char matri[15];
char nom[21];
char prenom[50];
float moy;
};
int main()
{
//************** Permet de créer un tableau de type étudiant**************

struct Etudiant etud[10];
int i;
//************** Permet remplir un tableau de type étudiant **************
for(i=0;i<9;i++)
{
printf("MATRICULE N° %d: ",i+1);gets(etud[i].matri);
printf("NOM N° %d: ",i+1);gets(etud[i].nom);
printf("PRENOM N° %d: ",i+1);gets(etud[i].prenom);
printf("MOYENNE N° %d: ",i+1);scanf("%f",&etud[i].moy);
}
//************** Permet de filtrer et afficher du tableau la liste des étudiants ayant une moyenne supérieure ou égale à 12 **************
for(i=0;i<9;i++)
{
if(etud[i].moy > 12)
{
printf("MATRICULE : %s ",etud[i].matri);
printf("NOM : %s ",etud[i].nom);
printf("PRENOM : %s ",etud[i].prenom);
printf("MOYENNE : %4.1f: ",etud[i].moy);
}
}

return 0;
}


```