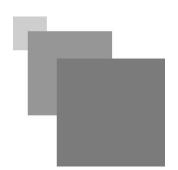
# LEÇON 3: LES ENREGISTREMENTS

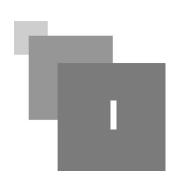
SANE ARNAUD



# Table des matières

I - 1- Structure d'enregistrement	3
II - 2- Syntaxe	4
III - Application 2:	6
IV - 3- Manipulation d'une structure	7
V - Application 3:	9
VI - 4 - Travaux dirigés	10

# 1- Structure d'enregistrement



#### 1

# Définition : 1.1- Définition

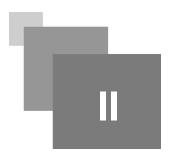
Un enregistrement, appelé structure En langage C, est une variable complexe qui permet de désigner sous un seul nom un ensemble de valeurs pouvant être de types différents.

### 1.2- Caractérisation

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

 $A vant \ de \ d\'eclarer \ une \ variable \ structure \ , \ il \ faut \ au \ pr\'ealable \ cr\'eer \ son \ type \ ainsi \ que \ les \ diff\'erents \ champs \ qui \ le \ compose.$ 

# 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 >; \\ \dots \\ \} :
```

La déclaration d'une structure est constituée de :

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

# 

### 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

## $\'{E}nonc\'{e}$ :

Déclarer les variables p1 et p2 de type structure de personne

### $R\'{e}solution$ :

```
1 CAS 1 :
2 struct Personne{
3 char nom[20];
4 char prenom[50];
5 int age;
6 char cel[15];
7 };
8 struct Personne p1,p2;
9 CAS 2 :
10 struct Personne{
11 char nom[20];
12 char prenom[50];
13 int age;
14 char cel[15];
15 }p1,p2;
16
```

# Application 2:



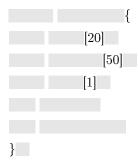
# Exercice 1

### 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



# 3- Manipulation d'une structure



## 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\!>;
```

# 

Nous allons utiliser la structure précédente dans notre exemple.

## 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";
```

# Figure Exemple

## Application:

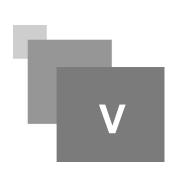
Écrire un programme qui permet de donner l'aîné de deux personnes saisies au préalable par l'utilisateur.

## $R\'{e}solution:$

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

```
4 char nom[21];
5 char prenom[50];
6 int age;
7 char cel[15];
8 };
9 int main()
10 {
11 struct Personne p1,p2;
12 //****** Permet de rentrer les informations de la première personne
13 printf("SAISIE LES INFORMATIONS DE LA PREMIERE PERSONNE\n");
14 printf("NOM: ");gets(p1.nom);
15 printf("PRENOM: ");gets(p1.prenom);
16 printf("AGE: ");scanf("%d",&p1.age);
17 printf("CEL: ");gets(p1.cel);
18 //****** Permet de rentrer les informations de la deuxième personne
  ******
19 printf("SAISIE LES INFORMATIONS DE LA DEUXIEME PERSONNE\n");
20 printf("NOM: ");gets(p2.nom);
21 printf("PRENOM: ");gets(p2.prenom);
22 printf("AGE: ");scanf("%d",&p2.age);
23 printf("CEL: ");gets(p2.cel);
*****
25 if(pl.age > p2.age)
26 {printf("NOM: %s PRENOM: %s AGE: %2d CEL: %s",pl.nom,pl.prenom,pl.age,pl.cel);
27 }
28 else
29 {
30 if (pl.age < p2.age)
31 {printf("NOM: %s PRENOM: %s AGE: %2d CEL: %s",p2.nom,p2.prenom,p2.age,p2.cel);
32 }
33 else
34 {
35 printf("Les deux ont le même âge ");
36 }
37 }
38
39 return 0;
40 }
```

# Application 3:



# 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:

```
1 #include <stdio.h>
 2 struct Etudiant
 4 char matri[15];
5 char nom[21];
6 char prenom[50];
7 float moy;
8 };
9 int main()
11 //****** Permet de créer un tableau de type étudiant**********
13 struct Etudiant etud[10];
14 int i;
15 //****** Permet remplir un tableau de type étudiant *********
16 for(i=0;i<9;i++)
18 printf("MATRICULE N° %d: ",i+1);gets(etud[i].matri);
19 printf("NOM N° %d: ",i+1);gets(etud[i].nom);
20 printf("PRENOM N° %d: ",i+1);gets(etud[i].prenom);
21 printf("MOYENNE N° %d: ",i+1);scanf("%f",&etud[i].moy);
23 //******* Permet de filtrer et afficher du tableau la liste des étudiants
  ayant une moyenne supérieure ou égale à 12 ***********
24 for(i=0;i<9;i++)
25 {
26 if(etud[i].moy > 12)
28 printf("MATRICULE : %s ",etud[i].matri);
29 printf("NOM : %s ",etud[i].nom);
30 printf("PRENOM : %s ",etud[i].prenom);
31 printf("MOYENNE : %4.1f: ",etud[i].moy);
32 }
33 }
35 return 0;
36 }
37
```