Les structures en C

Une structure est une variable qui est définie elle-même par plusieurs variables 😊

Exemple : une structure livre.

**Définition d’une structure**

Etant donné qu’une structure est une variable il faut comme toujours lui donner un nom et un type.

Ici en langage C ça donne 🡪 **struct** livre {

Ici on peut avoir plusieurs éléments :

* Une variable
* Une autre structure
* Des pointeurs

Char nomDuLivre[256] ;

Char nomAuteur[100] ;

Char editeur [100] ;

Int NbrPage ;

Char ISBN ;

} ;

Cette structure doit être créée en dehors du main() !

Exemple d’une structure dans une structure  :

#include <stdio.h>

struct n{

float points[5];

int ponderation ;};

struct eleve

{ char noms[25];

struct n notes;

};

int main()

{ struct eleve el;

int i,j,ok;

printf("\nNom : ");

scanf("%s",el.noms);

for(i=0;i<5;i++)

{

printf("\nNote %d : ",i+1);

scanf("%f",&el.notes.points[i]);

}

printf("\n\nRecap : \n");

printf("\nPour %s, voici les notes : \n",el.noms);

for(i=0;i<5;i++)

{

printf("\n%.2f",**el.notes.points[i]**);

}

La structure n doit avoir été définie avant celle qui la contient !!

**Création d’un élément de la structure.**

Cette fois dans le main() 😊

Il y a deux méthodes :

1. Soit l’initialiser par nous même dans le programme

Struct livre livre1 = {«Le Livre du C Premier Langage », « Claude Delannoy », « Eyrolles », 251, « 978-2212110524 »} ; *//On crée un livre1 qui a une structure livre*

1. Ou par demande à l’utilisateur :

Printf(« Quel est le nom du livre ? : ») ;

Scanf(« %s », **livre1.nomDuLivre**) ;

Printf(« Quel est le nom de l’auteur ? : ») ;

Scanf(« %s », **livre1. nomAuteur**) ;

Printf(« Quel est l’editeur ? : ») ;

Scanf(« %s », **livre1. editeur**) ;

Printf(« Quel est Nombre de pages ? : ») ;

Scanf(« %d », **livre1. NbrPage**) ;

Printf(« Quel est le numéro isbn ? : ») ;

Scanf(« %s », **livre1. ISBN**) ;

Pour afficher par exemple le nom du livre, le nom de l’auteur et le nombre de page du livre on fait :

Printf(« Le livre %s de l’auteur %s fait %d page(s) », livre1.nomDuLivre, livre1.nomAuteur, livre1.NbrPage) ;

**Création de nos propres type de données(genre int, float, …)**

Pour créer votre propre type de donnée exemple eleve on va utiliser le mot réservé ***TYPEDEF***

* Typedef struct Eleve{

Char nom[35] ;

Char prenom[35] ;

Int age ;

Char section [35] ;

} Etudiant;

Ceci veut dire que La structure Eleve a pour type Etudiant 😉

Dans la création d’un élément élevé on aura plus qu’à mettre 🡪 Etudiant eleve1 = {« Calens», « Alexis », …, « Informatique de gestion »}

**Modification d’une valeur de la structure.**

|  |  |
| --- | --- |
| Pour les float, int, … | Eleve1.age = 22 ; |
| Pour les char, chaine | Strcpy(eleve1.nom, « Callens ») ; |

**Utilisation des pointeurs avec les structures.**

On va donc créer une fonction qui va ajouter un étudiant  etu1:

En dehors du main on va donc ajouter 🡪

Void AjouterEtudiant(Etudiant \*pointeur) {

(

Strcpy( (\*pointeur).nom, « Curon ») ;

Strcpy( (\*pointeur).prenom, « Mick ») ;

(\*pointeur).age = 20 ;

Strcpy( (\*pointeur).section, « Informatique de gestion») ;

**Sans oublier le prototype** de la fonction d’AjoutEtudiant 🡪 On va pointer vers chaque élément de la structure qui forme l’étudiant. 🡪

void AjouterEtudiant(Etudiant \*pointeur);

}

Dans le main, on va appeler la fonction d’ajout d’étudiant et lui passer l’adresse de

AjouterEtudiant (&etu1) ; //Ajout de l’étudiant etu1 qui aura les adresses des pointeurs (de la fonction AjoutEtudiant)

Et ensuite afficher les valeurs présentent dans les adresses de la mémoire

printf("Nom de l'etudiant : %s", etu1.nom);

printf("\nPrenom de l'etudiant : %s", etu1.prenom);

printf("\nAge de l'etudiant : %d", etu1.age);

printf("\nSection de l'etudiant %s est %s", etu1.prenom, etu1.section);

**En programme final : Nous avons donc 🡪**

**Le code** https://mega.nz/#!VZ9FkAiJ!nwxdtLTP7q\_VmxFi0kTUqLYUPAr9GjVM7sW44lBk-7Q

#include <stdio.h>

#include <string.h>

typedef struct Eleve{

Création de notre type de donnée (Etudiant) qui aura comme structure Eleve

char nom[35] ;

char prenom[35] ;

int age ;

char section [35] ;

} Etudiant;

Prototype de la procédure d’ajout d’un étudiant

void AjouterEtudiant(Etudiant \*pointeur);

int main(){

On créé etu1 de type Etudiant

Etudiant etu1 ;

Appel de la procédure d’ajoutEtudiant en recevant le contenu des adresses

AjouterEtudiant(&etu1);

printf("Nom de l'etudiant : %s", etu1.nom);

printf("\nPrenom de l'etudiant : %s", etu1.prenom);

printf("\nAge de l'etudiant : %d", etu1.age);

printf("\nSection de l'etudiant %s est %s", etu1.prenom, etu1.section);

return 0;

}

void AjouterEtudiant(Etudiant \*pointeur){

Création de la procédure d’AjoutEtudiant dont le pointeur \*pointeur pointe vers les adresses contenant les variables.

strcpy((\*pointeur).nom,"Curon") ;

strcpy((\*pointeur).prenom,"Mick") ;

(\*pointeur).age = 20 ;

strcpy((\*pointeur).section,"Informatique de gestion") ;

}