Les structures en C

-

1. ## Objectifs

2. ## Introduction

   - Une structure est une suite finie d’objets de types différents.
   - Les variables d'une structure sont appelées membres dont chacune est distingué par un identificateur et peut avoir n'importe quel type, y compris des tableaux ou d'autres structures.
   - Contrairement aux tableaux, les différents éléments d’une structure n’occupent pas nécessairement des zones contigües en mémoire

3. ## Syntaxe des Structures en C

   - En C, les structures permettent de regrouper plusieurs variables de différents types sous un même nom. Elles sont particulièrement utiles pour représenter des enregistrements complexes comme des informations d'étudiants, des produits, des dates, etc.
   - Voici la syntaxe générale pour définir une structure et des exemples pour illustrer cela.

     1. **Définition Générale d'une Structure**

```
struct nom_modele {
    Type1 nom_membre1;
    Type2 nom_membre2;
    ...
    TypeN nom_membreN;
};
```

     2. **Exemples de Structures**

        1. **Structure Produit**

apcpedagogie.com

```
struct Produit {
    char libelle[30];
    float prix;
};
```

2. **Structure date**

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

3. **Structure etudiant Utilisant une Structure date**

- Pour illustrer l'utilisation d'une structure à l'intérieur d'une autre structure, voici la structure etudiant qui inclut une structure date pour représenter la date de naissance :

```
struct etudiant {
    long matricule;
    char nom[20];
    char prenom[20];
    struct date date_naissance;
    float moyenne;
};
```

4. **Programme Complet : Gestion des Étudiants**

- Ce programme demande la taille d’un tableau d’étudiants à créer, saisit les éléments de chaque étudiant au clavier, les affiche, affiche leur somme des moyennes et affiche l’étudiant ayant la moyenne la plus élevée.

5. **Définition des Structures et des Fonctions**

apcpedagogie.com

```
#include
#include

// Définition de la structure date
typedef struct date {
    int jour;
    int mois;
    int annee;
} date;

// Définition de la structure etudiant
typedef struct etudiant {
    char nom[20];
    char prenom[20];
    char mat[10];
    float *Tmoy;
    float moyg;
    date date_naissance;
} etudiant;

// Fonction pour calculer la moyenne générale des notes
float calcul_moyg(float *Tmoy, int n) {
    float somme = 0.0;
    for (int i = 0; i < n; i++) {
        somme += Tmoy[i];
    }
    return (n > 0) ? (somme / n) : 0.0;
}

// Fonction pour saisir les informations d'un étudiant
etudiant saisie_etudiant() {
    etudiant etu;
    int nb_matieres;

    // Saisie des informations de l'étudiant
    printf("Nom de l'étudiant: ");
    scanf("%s", etu.nom);

    printf("Prénom de l'étudiant: ");
    scanf("%s", etu.prenom);

    printf("Matricule de l'étudiant: ");
```

apcpedagogie.com

```
    scanf("%s", etu.mat);

    printf("Date de naissance (jour mois annee): ");
    scanf("%d %d %d", &etu.date_naissance.jour,
&etu.date_naissance.mois, &etu.date_naissance.annee);

    // Saisie du nombre de matières
    printf("Nombre de matières: ");
    scanf("%d", &nb_matieres);

    // Allocation dynamique de la mémoire pour les notes
    etu.Tmoy = (float *)malloc(nb_matieres * sizeof(float));
    if (etu.Tmoy == NULL) {
        printf("Échec de l'allocation mémoire.\n");
        exit(1);
    }

    // Saisie des notes
    printf("Saisir les notes:\n");
    for (int i = 0; i < nb_matieres; i++) {
        printf("Note %d: ", i + 1);
        scanf("%f", &etu.Tmoy[i]);
    }

    // Calcul de la moyenne générale
    etu.moyg = calcul_moyg(etu.Tmoy, nb_matieres);

    return etu;
}
```

6. **Programme Principal**

```
int main() {
    int nb_etudiants;
    etudiant *classe;
    float somme_moyennes = 0.0, moyenne_classe;
    int indice_meilleure_moyenne = 0;

    // Demande du nombre d'étudiants
    printf("Nombre d'étudiants: ");
    scanf("%d", &nb_etudiants);
```

apcpedagogie.com

```
    // Allocation dynamique de mémoire pour le tableau d'étudiants
    classe = (etudiant *)malloc(nb_etudiants * sizeof(etudiant));
    if (classe == NULL) {
        printf("Échec de l'allocation mémoire.\n");
        return 1;
    }

    // Saisie des informations de chaque étudiant et calcul de la
moyenne
    for (int i = 0; i < nb_etudiants; i++) {
        printf("\nSaisie des informations pour l'étudiant %d:\n", i +
1);
        classe[i] = saisie_etudiant();
        somme_moyennes += classe[i].moyg;
        if (classe[i].moyg > classe[indice_meilleure_moyenne].moyg) {
            indice_meilleure_moyenne = i;
        }
    }

    // Calcul de la moyenne de la classe
    moyenne_classe = somme_moyennes / nb_etudiants;

    // Affichage des résultats
    printf("\nMoyenne de la classe: %.2f\n", moyenne_classe);
    printf("Étudiant avec la meilleure moyenne:\n");
    printf("Matricule: %s\n", classe[indice_meilleure_moyenne].mat);
    printf("Nom: %s\n", classe[indice_meilleure_moyenne].nom);
    printf("Prénom: %s\n", classe[indice_meilleure_moyenne].prenom);
    printf("Moyenne Générale: %.2f\n",
classe[indice_meilleure_moyenne].moyg);

    // Libération de la mémoire allouée pour les notes de chaque
étudiant et pour le tableau d'étudiants
    for (int i = 0; i < nb_etudiants; i++) {
        free(classe[i].Tmoy);
    }
    free(classe);

    return 0;
}
```

apcpedagogie.com

7. **Explications**

    1. **Déclarations et Allocations :**
        - La structure date est définie pour stocker les dates de naissance.
        - La structure etudiant inclut un champ date_naissance de type date.
    2. **Fonction `calcul_moyg`:**
        - Calcule la moyenne des notes stockées dans un tableau dynamique de float.
    3. **Fonction saisie_etudiant :**
        - Saisit les informations de l'étudiant, alloue dynamiquement la mémoire pour les notes, les remplit et calcule la moyenne générale.
    4. **Programme Principal :**
        - Demande le nombre d'étudiants et alloue dynamiquement un tableau d'étudiants.
        - Remplit les informations de chaque étudiant, calcule la moyenne de la classe et identifie l'étudiant ayant la meilleure moyenne.
        - Affiche les résultats et libère la mémoire allouée dynamiquement.
        - Ce programme illustre la gestion des structures et de la mémoire dynamique en C, en manipulant des enregistrements complexes et en assurant une gestion correcte de la mémoire.

4. Création d'Instance (Objet) en C

- En langage C, pour déclarer une instance (ou objet) d'une structure, vous devez suivre certaines syntaxes.

1. **Déclaration d'une Structure et Instances**

    1. **Déclaration d'une structure et d'instances de manière séparée :**

apcpedagogie.com

```
// Déclaration de la structure
struct etudiant {
    long matricule;
    char nom[20];
    char prenom[20];
    struct date {
        int jour;
        int mois;
        int annee;
    } date_naissance;
    float moyenne;
};

// Déclaration d'instances de la structure
struct etudiant E1;
struct etudiant classe[20]; // Tableau de 20 étudiants
```

2. **Déclaration et définition d'instances dans la même instruction :**
   - Si le modèle de la structure n'a pas été déclaré précédemment, vous pouvez le faire en une seule instruction :

```
// Déclaration de la structure et instances en même temps
struct livre {
    char nom[40];
    char auteur[40];
    int isbn;
} livre1, livre2;
```

2. **Exemples**

1. **Exemple 1 : Gestion de Livres**
   - Définissons une structure livre et déclarons deux instances livre1 et livre2 :

```
#include
```

apcpedagogie.com

```
// Déclaration de la structure et instances
struct livre {
    char nom[40];
    char auteur[40];
    int isbn;
} livre1, livre2;

int main() {
    // Initialisation des instances
    strcpy(livre1.nom, "Le Petit Prince");
    strcpy(livre1.auteur, "Antoine de Saint-Exupéry");
    livre1.isbn = 9782070612758;

    strcpy(livre2.nom, "1984");
    strcpy(livre2.auteur, "George Orwell");
    livre2.isbn = 9780451524935;

    // Affichage des informations des livres
    printf("Livre 1 :\n");
    printf("Nom : %s\n", livre1.nom);
    printf("Auteur : %s\n", livre1.auteur);
    printf("ISBN : %d\n", livre1.isbn);

    printf("\nLivre 2 :\n");
    printf("Nom : %s\n", livre2.nom);
    printf("Auteur : %s\n", livre2.auteur);
    printf("ISBN : %d\n", livre2.isbn);

    return 0;
}
```

2. **Exemple 2 : Gestion d'Étudiants**

   - Définissons une structure etudiant et déclarons une instance E1 et un tableau de 20 étudiants classe :

```
#include
#include

// Déclaration de la structure etudiant
struct date {
```

apcpedagogie.com

```
    int jour;
    int mois;
    int annee;
};

struct etudiant {
    long matricule;
    char nom[20];
    char prenom[20];
    struct date date_naissance;
    float moyenne;
};

// Déclaration d'une instance et d'un tableau d'étudiants
struct etudiant E1;
struct etudiant classe[20];

int main() {
    // Initialisation de l'instance E1
    E1.matricule = 123456;
    strcpy(E1.nom, "Dupont");
    strcpy(E1.prenom, "Jean");
    E1.date_naissance.jour = 15;
    E1.date_naissance.mois = 4;
    E1.date_naissance.annee = 1995;
    E1.moyenne = 14.5;

    // Affichage des informations de l'étudiant E1
    printf("Etudiant E1 :\n");
    printf("Matricule : %ld\n", E1.matricule);
    printf("Nom : %s\n", E1.nom);
    printf("Prenom : %s\n", E1.prenom);
    printf("Date de naissance : %d/%d/%d\n",
E1.date_naissance.jour, E1.date_naissance.mois,
E1.date_naissance.annee);
    printf("Moyenne : %.2f\n", E1.moyenne);

    return 0;
}
```

apcpedagogie.com

5. Initialisation d'un Objet en C

- L'initialisation d'un objet de type structure peut se faire directement lors de sa déclaration en utilisant une syntaxe spécifique.

1. **Initialisation Simple**

1. **Exemple avec une structure simple date :**

```
#include

// Déclaration de la structure date
struct date {
    int jour;
    int mois;
    int annee;
};

int main() {
    // Initialisation de l'objet aujourd'hui
    struct date aujourd'hui = {30, 4, 2018};

    // Affichage des valeurs de l'objet aujourd'hui
    printf("Date: %d/%d/%d\n", aujourd'hui.jour, aujourd'hui.mois,
aujourd'hui.annee);

    return 0;
}
```

2. **Exemple avec une structure simple livre :**

```
#include

// Déclaration de la structure livre
struct livre {
    char nom[40];
    char auteur[40];
    int isbn;
```

apcpedagogie.com

```
};

int main() {
    // Initialisation de l'objet livre1
    struct livre livre1 = {"L'alchimiste", "Paulo Coelho",
21673874};

    // Affichage des valeurs de l'objet livre1
    printf("Livre:\n");
    printf("Nom: %s\n", livre1.nom);
    printf("Auteur: %s\n", livre1.auteur);
    printf("ISBN: %d\n", livre1.isbn);

    return 0;
}
```

2. **Initialisation d'une Structure dans une Structure**

1. **Exemple avec une structure etudiant contenant une structure date :**

```
#include
#include

// Déclaration de la structure date
struct date {
    int jour;
    int mois;
    int annee;
};

// Déclaration de la structure etudiant
struct etudiant {
    long matricule;
    char nom[20];
    char prenom[20];
    struct date date_naissance;
    float moyenne;
};

int main() {
```

apcpedagogie.com

```
    // Initialisation de l'objet étudiant m
    struct etudiant m = {132453, "Mohamed", "ben amor", {12, 4,
1995}, 12.65};

    // Affichage des valeurs de l'objet m
    printf("Etudiant:\n");
    printf("Matricule: %ld\n", m.matricule);
    printf("Nom: %s\n", m.nom);
    printf("Prenom: %s\n", m.prenom);
    printf("Date de Naissance: %d/%d/%d\n", m.date_naissance.jour,
m.date_naissance.mois, m.date_naissance.annee);
    printf("Moyenne: %.2f\n", m.moyenne);

    return 0;
}
```

6. Accès aux Membres d'une Structure en C

- Pour accéder aux membres d'une structure en C, on utilise l'opérateur . (point). Cet opérateur permet de désigner un membre spécifique d'un objet de structure.

1. **Exemple 1 : Calcul de la Norme d'un Nombre Complexe**

   - Voici un exemple de programme qui calcule la norme d'un nombre complexe :

```
#include
#include

// Déclaration de la structure complexe
struct complexe {
    float relle;
    float img;
};

int main() {
    // Déclaration et initialisation de l'objet z
    struct complexe z;
    z.relle = 5.3;
    z.img = 12.4;
```

apcpedagogie.com

```
    // Calcul de la norme
    float norme = sqrt(z.relle * z.relle + z.img * z.img);

    // Affichage de la norme
    printf("La norme du nombre complexe est: %.2f\n", norme);

    return 0;
}
```

2. **Exemple 2 : Affichage des Étudiants d'une Classe**

   - Supposons que nous ayons une structure etudiant et que nous souhaitons afficher les détails des étudiants d'une classe de 10 étudiants déjà initialisés :

```
#include
#include

// Déclaration de la structure date
struct date {
    int jour;
    int mois;
    int annee;
};

// Déclaration de la structure etudiant
struct etudiant {
    char nom[20];
    char prenom[20];
    struct date date_naissance;
};

// Déclaration et initialisation d'un tableau de 10 étudiants
struct etudiant cl[10] = {
    {"Dupont", "Jean", {15, 4, 1995}},
    {"Martin", "Alice", {8, 6, 1996}},
    {"Durand", "Marie", {23, 12, 1994}},
    {"Petit", "Paul", {5, 11, 1997}},
    {"Moreau", "Sophie", {30, 1, 1998}},
    {"Lefevre", "Louis", {14, 9, 1995}},
    {"Leroy", "Nathalie", {2, 3, 1996}},
```

apcpedagogie.com

```
    {"Roux", "Jacques", {10, 7, 1994}},
    {"David", "Clara", {21, 10, 1997}},
    {"Bertrand", "Lucas", {17, 5, 1995}}
};

int main() {
    // Affichage des étudiants de la classe
    for(int i = 0; i < 10; i++) {
        printf("Nom: %s \t Prenom: %s \t Date de Naissance: %d/%d/%d\n",
               cl[i].nom, cl[i].prenom,
               cl[i].date_naissance.jour, cl[i].date_naissance.mois,
cl[i].date_naissance.annee);
    }

    return 0;
}
```

7. Applications

1. **Exercice 1**

   - Déclarer une variable P ayant les informations suivantes:
     − Nom (chaîne de caractères) − Prénom (chaîne de caractères) − Date de naissance: Jour (entier), Mois (entier) et Année (entier) − Matricule (chaîne de caractères)
   - Remplir P avec les informations suivantes: Nom: "Amer" Prénom: "Salem" Date de
   - naissance: "03/04/2005" Matricule: "A32"

2. **Exercice 2**

   - Soit une fiche employé (nom, prenom, matricule, salaire)
   - Saisir un tableau de 100 employés puis afficher le nom, le prénom et la matricule de l’employé ayant le salaire le plus élévé.

3. **Exercice 3**

   - Sachant qu’un nombre complexe est composé de partie réelle et de partie imaginaire,
   - créer la structure Complexe puis saisir deux nombres complexes C1 et C2 et afficher leur sommes et leur produit.

apcpedagogie.com

Cours en C

Source : https://apcpedagogie.com/les-structures-en-c · téléchargé le 08/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com