

## TP n°4 – Les Structures

Lorsqu'une absence est constatée, le service scolarité en garde une trace en enregistrant pour l'élève concerné :

- Son nom,
- Son prénom,
- Son année,
- La date de l'absence,
- La durée de l'absence,
- Le motif.

### Exercice :

Dans cet exercice, vous utiliserez un tableau de taille 100 pour stocker les absences. Un menu dans le programme principal permettra à l'utilisateur d'effectuer les actions des questions 2 à 6 :

**Question 1 :** Trouver en C le type des données adaptées au stockage de ces informations.

**Question 2 :** Écrire en C une fonction permettant la saisie d'une nouvelle absence.

**Question 3 :** Écrire en C une fonction pour déterminer la durée moyenne des absences.

**Question 4 :** Écrire en C une fonction listant les absences d'une année donnée.

**Question 5 :** Écrire une fonction listant les absences d'un élève donné. Donner le nombre d'absences relevées pour cet élève.

**Question 6 :** Permettre la suppression d'une absence pour un élève à une date donnée. La case du tableau où était l'absence ne doit pas rester vide.

### Corrigé :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#define MAXI 100
```

```
struct Date
{
    int jour;
    int mois;
    int annee;
```

```

};

struct Absence
{
    char nom[30];
    char prenom[30];
    int annee;
    struct Date date_abs;
    int duree;
    char motif[50];
};

void saisie(struct Absence tab[], int place)
{
    printf("Nom de l'eleve : ");
    __fpurge(stdin);
    scanf("%s", tab[place].nom);
    printf("Prenom de l'eleve : ");
    __fpurge(stdin);
    scanf("%s", tab[place].prenom);
    printf("Annee de l'eleve : ");
    scanf("%d", & tab[place].annee);
    printf("Jour de l'absence : ");
    scanf("%d", & tab[place].date_abs.jour);
    printf("Mois de l'absence : ");
    scanf("%d", & tab[place].date_abs.mois);
    printf("Annee de l'absence : ");
    scanf("%d", & tab[place].date_abs.annee);
    printf("Duree de l'absence : ");
    scanf("%d", & tab[place].duree);
    printf("Motif de l'absence : ");
    __fpurge(stdin);
    fgets(tab[place].motif, 50, stdin);
}

float moyenne(struct Absence tab[], int place)
{
    float somme = 0;
    int i;

    for(i=0; i<place; i++)
        somme += tab[i].duree;
    if(place != 0)
        return somme / place;
    else
        return 0;
}

void liste_annee(struct Absence tab[], int place)
{
    int i;
    int an;
    printf("De quel annee souhaitez-vous les absences : ");
    scanf("%d", &an);
}

```

```

    for(i=0; i<place; i++)
    {
        if(tab[i].date_abs.annee == an)
            printf(" - %s %s absent le %d/%d/%d pendant %d jour(s)\n",
                tab[i].nom, tab[i].prenom, tab[i].date_abs.jour,
                tab[i].date_abs.mois, tab[i].date_abs.annee, tab[i].duree);
    }
}

void liste_eleve(struct Absence tab[], int place)
{
    char nom[30];
    int i;

    __fpurge(stdin);
    printf("Entrez le nom de l'eleve : ");
    scanf("%s", nom);

    for(i=0; i<place; i++)
    {
        if (strcmp(tab[i].nom, nom) == 0)
            printf(" - %s %s absent le %d/%d/%d pendant %d jour(s)\n",
                tab[i].nom, tab[i].prenom, tab[i].date_abs.jour,
                tab[i].date_abs.mois, tab[i].date_abs.annee, tab[i].duree);
    }
}

int supprimer_absence(struct Absence tab[], int place)
{
    int jour, mois, annee;
    char nom[30];
    int i = 0;
    int trouve = 0;

    __fpurge(stdin);
    printf("Entrez le nom de l'eleve : ");
    scanf("%s", nom);
    printf("Entrez le jour de l'absence : ");
    scanf("%d", &jour);
    printf("Entrez le mois de l'absence : ");
    scanf("%d", &mois);
    printf("Entrez l'annee de l'absence : ");
    scanf("%d", &annee);

    while(i < place && trouve == 0)
    {
        if(strcmp(tab[i].nom, nom) == 0 && tab[i].date_abs.jour == jour &&
            tab[i].date_abs.mois == mois && tab[i].date_abs.annee == annee)
        {
            if(i < place-1)
            {
                // on recopie le dernier enregistrement à la place
                strcpy(tab[i].nom, tab[place-1].nom);
                strcpy(tab[i].prenom, tab[place-1].prenom);
                tab[i].annee = tab[place-1].annee;
            }
        }
    }
}

```

```

        tab[i].date_abs.jour = tab[place-1].date_abs.jour;
        tab[i].date_abs.mois = tab[place-1].date_abs.mois;
        tab[i].date_abs.annee = tab[place-1].date_abs.annee;
        tab[i].duree = tab[place-1].duree;
        strcpy(tab[i].motif, tab[place-1].motif);
    }
    trouve = 1;
}
i++;
}
return trouve;
}

int main()
{
    int choix;
    int indice = 0; // Indice du tableau où sera saisie la prochaine absence
    struct Absence liste[MAXI];

    printf("Programme de gestion des absences\n\n");

    do
    {
        printf("Que voulez-vous faire :\n");
        printf("1 - Saisir une absence\n2 - Duree moyenne\n3 - Absence d'une\n\n
        annee\n4 - Absences d'un eleve\n5 - Supprimer une absence\n6\n\n
        - Quitter\n\nVotre choix : ");

        scanf("%d", &choix);

        switch(choix)
        {
            case 1:
                if(indice == MAXI)
                    printf("Impossible d'ajouter une absence");
                else
                {
                    saisie(liste, indice);
                    indice++;
                }
                break;
            case 2:
                printf("moyenne : %f\n", moyenne(liste, indice));
                break;
            case 3:
                liste_annee(liste, indice);
                break;
            case 4:
                liste_eleve(liste, indice);
                break;
            case 5:
                if(supprimer_absence(liste, indice) == 1)
                {
                    indice--;
                    printf("Absence supprimee\n");
                }
            }
        }
    } while(choix != 6);
}

```

```
    }  
    else  
        printf("Absence non trouvee\n");  
        break;  
  
    case 6:  
  
        printf("Fin du programme\n");  
        break;  
  
    default: printf("Choix non valide\n");  
  
    }  
}while(choix != 6);  
return 0;  
}
```