

TP 4 - IGI 2101

Chaîne de caractères, structures

Exercice 1 : structures, chaîne de caractères

- 1) Dans un programme "TP4ex1" : avant le main, déclarez une structure "Eleve" (sans accent !), qui comportera les champs :
 - a. **Nom** : chaîne de 30 caractères maximum
 - b. **Prenom** : chaîne de 30 caractères maximum
 - c. **Age** : un entier (int)
- 2) Dans le main, allouer un tableau, que vous appellerez **ElevesTab** , de 5 éléments de structure **Eleve**. Il ne sera pas initialisé lors de la déclaration.
- 3) Ecrire une boucle qui demande de saisir le Nom, Prenom, age de chaque élève du tableau. **Pour cela il faut utiliser la fonction scanf présentée dans l'encadré à la fin du sujet puisque nous n'avons pas eu le temps de la voir en cours).**
- 4) Ecrire une boucle qui permet d'afficher les champs précédemment saisis pour tous les élèves du tableau.

Exercice 2 : modification de l'âge

Dupliquez le programme précédent pour faire un nouveau programme "TP4ex2.c" (commande Unix « cp »). Modifier ce nouveau programme comme expliqué ci-dessous.

Après l'affichage des noms, le programme doit demander à l'utilisateur de saisir l'un des noms stocké dans le tableau, afin de pouvoir modifier son âge. Pour cela il faut :

- a. Demander de saisir le nom à rechercher,
- b. Dans une boucle (« for » ou « while »), comparer le nom qui a été saisi avec chacun des noms du tableau (attention nous avons vu que l'on ne pouvait pas utiliser l'opérateur égal = pour comparer des chaînes de caractères !). Si le nom est trouvé alors on demande de saisir une nouvelle valeur pour l'âge (c'est-à-dire utiliser un scanf), suivi d'un break pour quitter la boucle.
- c. On affiche à nouveau l'ensemble des éléments du tableau.

N.B. : lors de la mise au point et pour éviter de ressaisir plusieurs fois les données, vous pouvez initialiser les champs dès la déclaration.

Exercice 3 : tri de tableau d'entiers

Afin de vous préparer à l'exercice 4, écrire un programme qui va trier le tableau d'entiers suivant puis l'afficher.

```
int Tab[] = { 10, 4, 45, 453, 2, 4356, 23, 98, 23, 1 } ;
```

Vous pouvez utiliser l'algorithme de tri que vous souhaitez, par exemple le tri à bulle qui est très bien illustré avec la vidéo que vous pouvez trouver avec la recherche «bubble sort dance ».

Exercice 4 : tri de tableau de chaînes de caractères

Dupliquez le programme de l'exercice 1 (pas le 2 !) pour faire un nouveau programme "TP4ex3.c"
Modifier ce nouveau programme de la façon suivante

- a. Déclarer un second tableau *ElevesTabTri*, *non initialisé* de même taille et type qu'*ElevesTab*
- b. Après la saisie des champs faite en dans l'exercice 1-3), recopier le contenu de *ElevesTab* un à un dans *ElevesTabTri* à l'aide d'une boucle,
- c. Afficher le contenu de *ElevesTabTri* (pour vérifier que la copie de tableau s'est bien déroulée)
- d. **Trier** le tableau dans l'ordre alphabétique des noms.
- e. Afficher le contenu du tableau *ElevesTabTri* qui doit être trié si tout s'est bien passé

N.B. : lors de la mise au point et pour éviter de ressaisir plusieurs fois les données, vous pouvez initialiser les champs dès la déclaration.

Annexe : Fonction scanf

Cette fonction permet à l'utilisateur de saisir une variable ou une chaîne de caractères. Nous verrons son fonctionnement détaillé en cours.

Exemple pour faire saisir une variable entière **var1** déclarée par *int var1* ;

- `printf("Veuillez saisir la valeur que vous souhaitez stocker dans var1 : ");`
- `scanf("%d", &var1);` // le nombre entier qui sera saisi sera stocké à l'adresse de var1
- `printf("Maintenant var1=%d", var1);`

Exemple pour faire saisir une chaîne de caractères **mot** déclarée par *char mot[30]* ;

- `printf("Veuillez saisir un mot : ");`
- `scanf("%s", mot);` //mot est bien une adresse, l'adresse du 1^{er} élément du tableau
- `printf("le mot saisi est %s ");`

Le principe reste le même si nous utilisons des structures. Par exemple si **var1** et **mot** sont des champs d'une structure nous pourrions écrire :

```
struct {
```

```
int var1 ;  
  
char mot[30] ;  
  
} S1 ;
```

- `scanf("%d", &S1.var1);` //pour donner à scanf l'adresse du champs var1
- `scanf("%s", S1.mot);` //pour donner à scanf l'adresse du tableau mot