



Atelier de Programmation ***** TP2

Objectifs:

- Manipuler les types composés.
- Connaître les opérations de lecture et d'écriture dans un fichier texte en C
- Utiliser des fonctions de traitement de chaînes de caractères
- Manipuler l'allocation dynamique de mémoire
- Utilisation des pointeurs simples et doubles

Enoncé

Un étudiant est défini par son *nom*, son *prénom*, sa *matricule* numérique et sa *date de naissance* (jour/mois/année)

N.B: Les chaînes de caractères Nom et Prénom occupent dans la mémoire seulement le nombre nécessaire de caractères.

1- Définir le type composé **DateNaissance**

2- Définir le type composé **Etudiant**

3- Ecrire une fonction en C permettant d'enregistrer un tableau d'étudiants dans un fichier texte dont le chemin est passé en paramètre. Le nombre d'étudiants enregistrés est sauvegardé dans un fichier "info.txt".

4- Ecrire une fonction en C permettant de remplir un tableau par les étudiants enregistrés dans un fichier texte. Donner la fonction correspondante à chaque schéma en tenant compte que le nombre d'étudiants est enregistré dans le fichier "info.txt" et ceci permet d'allouer seulement la mémoire nécessaire pour le chargement des données.

Schéma 1 :

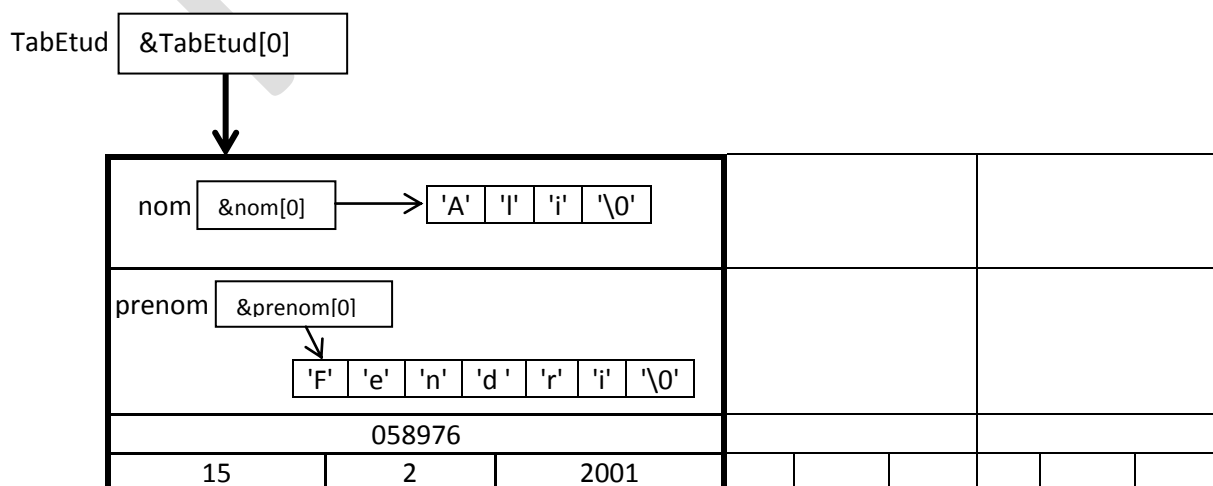


Schéma 2 :

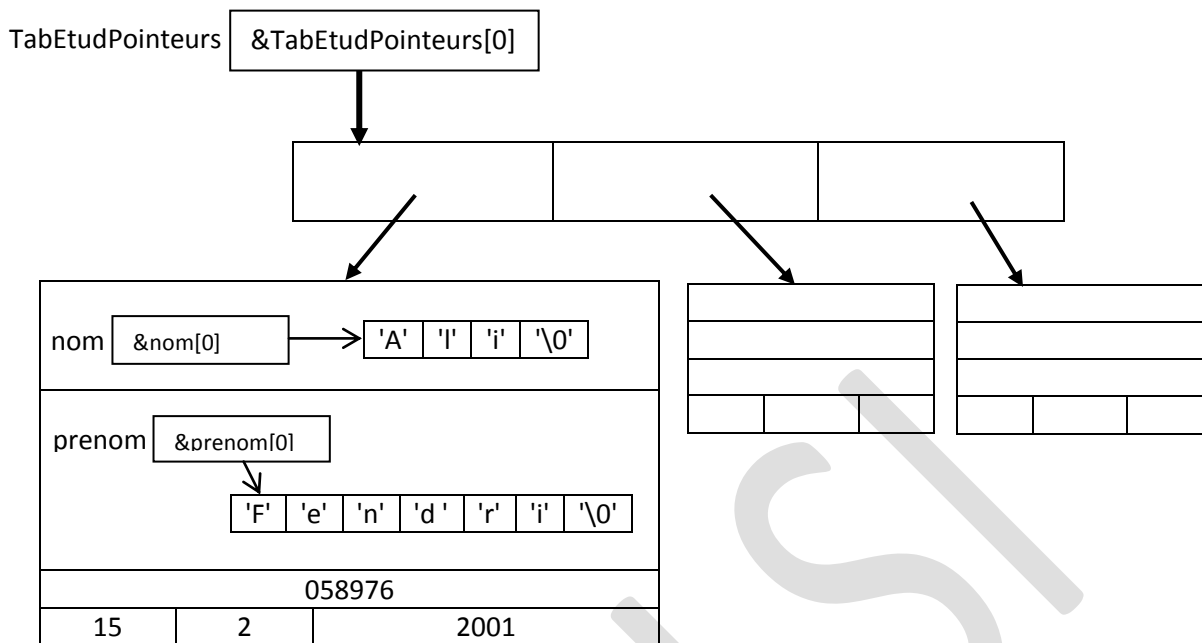
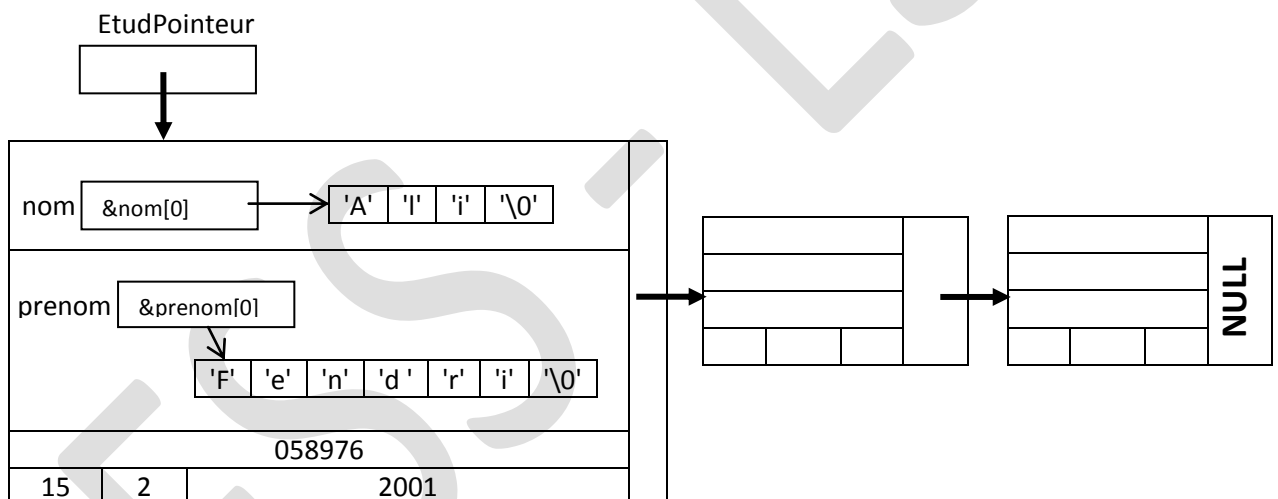


Schéma 3 :



5- On souhaite trier un tableau d'étudiants suivant l'ordre croissant des noms.

- Discuter le schéma adéquat (parmi les schémas ci dessus) pour une opération de tri.
- Implémenter la fonction de tri en considérant le schéma le plus adéquat.
- Implémenter la fonction de tri insertion en utilisant un **paramètre fonctionnel** définissant la relation d'ordre utilisée pour comparer deux personnes. En fait, la comparaison entre deux personnes peut être basée sur les noms, les prénoms ou l'âge.

Exemple de paramètre fonctionnel :

```
#include<stdio.h>

int addition(int x, int y)
{
    return x+y;
}

int multiplication(int x, int y)
{
    return x*y;
}

int soustraction(int x,int y)
{
    return x-y;
}

int calculer(int a, int b, int (*oper)(int,int))
{
    return (*oper) (a,b);
}

void main()
{
    int a, b;
    printf("a=");
    scanf("%d",&a);
    printf("b=");
    scanf("%d",&b);
    printf("%d\n",calculer(a,b,addition));
    printf("%d\n",calculer(a,b,multiplication));
    printf("%d\n",calculer(a,b,soustraction));
}
```

Complément de cours

Chaînes de caractères

1. Déclaration

Une chaîne est un tableau de caractères dont le dernier élément significatif est le caractère '\0'. Il faut donc différencier la longueur maximale de la chaîne (c'est à dire la dimension donnée lors de la définition) de sa longueur réelle (c'est à dire le nombre de caractères jusqu'au '\0').

char nom[dim]

Exemple :

```
char texte[dim];
```

Le compilateur réserve (dim-1) places en mémoire pour la chaîne de caractères. En effet, il ajoute toujours le caractère NUL ('\0') à la fin de la chaîne en mémoire.

2. Initialisation

```
char chaine[] = "Hello";
```

H	e	l	l	o	\0
---	---	---	---	---	----

```
char chaine[6] = "Hello";
```

H	e	l	l	o	\0
---	---	---	---	---	----

```
char chaine[8] = "Hello";
```

H	e	l	l	o	\0	0	0
---	---	---	---	---	----	---	---

```
char chaine[5] = "Hello";
```

H	e	l	l	o
---	---	---	---	---

==> Erreur à la compilation

3. Affichage

On peut utiliser la fonction **printf** avec le format **%s** :

```
char texte[10] = "BONJOUR";
```

```
printf("Voici le texte: %s\n", texte);
```

On peut aussi utiliser la fonction **puts** de la bibliothèque `<string.h>` sur :

```
puts(texte);                   ≡       printf("%s\n",texte);
```

4. Saisie

On peut utiliser la fonction **scanf** avec le format le format **%s**. On n'écrit pas le symbole **&**.
On peut aussi utiliser la fonction **gets** de la bibliothèque `<string.h>`.

```
char texte[10];
```

```
printf("entrer un texte: ");
```

```
scanf("%s", texte);                   ≡       gets(texte);
```

Remarques :

- La fonction **scanf** ne permet pas la saisie d'une chaîne comportant des espaces : les caractères saisis à partir de l'espace ne sont pas pris en compte mais rangés dans le tampon d'entrée. Pour saisir une chaîne comportant des espaces, il faut utiliser la fonction **gets**.
- A l'issue de la saisie d'une chaîne de caractères, le compilateur ajoute '\0' en mémoire après le dernier caractère.

Exemples :

```
char lieu[25];
```

```
int jour, mois, annee;
printf("Entrez lieu et date de naissance :");
scanf("%s %d %d %d", lieu, &jour, &mois, &annee);
```

5. Tableau de chaînes de caractères

Les tableaux de chaînes de caractères sont des tableaux de tableaux de caractères.

Exemple :

```
char tab[5][10] = { "zero", "un", "deux", "trois", "quatre" };
```

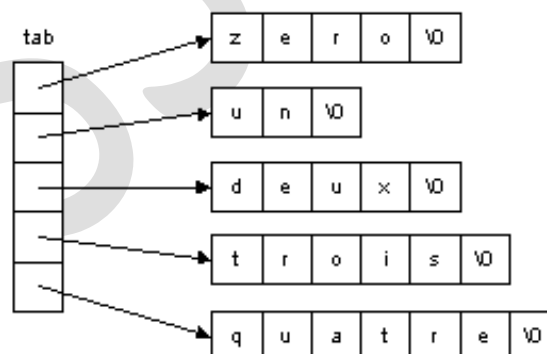
Ce qui donne comme implémentation :



On constate que le tableau contient de l'espace inutilisé relativement important (près de la moitié). Pour minimiser la mémoire nécessaire au stockage des chaînes, on peut déclarer **tab** comme un tableau de pointeurs :

```
char *tab[5] = { "zero", "un", "deux", "trois", "quatre" };
```

Ce qui donne comme implémentation :



6. Fonctions permettant la manipulation des chaînes

Les bibliothèques fournies avec les compilateurs contiennent de nombreuses fonctions de traitement des chaînes de caractères. Elles appartiennent à la bibliothèque `string.h`.

`string.h`:

<code>strlen(char* s)</code>	Fournit la longueur de la chaîne sans compter le '\0' final
<code>strcpy(char *s, char *t)</code>	Copie <t> vers <s>
<code>strcat(char* s, char *t)</code>	Ajoute <t> à la fin de <s>
<code>strcmp(char* s, char* t)</code>	Compare <s> et <t> lexicographiquement et fournit un résultat : négatif si <s> précède <t> zéro si <s> est égal à <t> positif si <s> suit <t>
<code>strncpy(char *s, char *t, int n)</code>	Copie au plus <n> caractères de <t> vers <s>
<code>strncat(char *s, char *t, int n)</code>	Ajoute au plus <n> caractères de <t> à la fin de <s>
<code>strchr(char *s, char c)</code>	Trouve la première occurrence du caractère c dans la chaîne s, c'est à dire la première fois où il apparaît
<code>strrev (char *s)</code>	Inversion de l'ordre de la chaîne s
<code>strupr (char *s)</code>	Conversion minuscule-majuscule de la chaîne s

Références

- <https://www.ltam.lu/cours-c/prg-c73.htm>
- <https://openclassrooms.com/fr/courses/19980-apprenez-a-programmer-en-c/15765-les-chaines-de-caracteres>