

TD 3 : Tableaux et chaînes de caractères

Cette feuille de TD a pour but l'apprentissage des tableaux et des chaînes de caractères. Nous apprendrons leur manipulation en faisant référence au traitement de file vu à la séance précédente.

Dans votre répertoire nommé TDc/, créez un sous-répertoire TDc/td3/ (avec la commande `mkdir` dans un terminal). Positionnez-vous dans ce sous-répertoire (avec la commande `cd`), vous y resterez toute cette séance.

1 Tableaux

Un tableau est une suite de zones mémoires adjacentes et de même type. D'un point de vue algorithmique, un tableau est donc une file, dont la classification dépend du contenu de celui-ci (on illustrera particulièrement ceci sur les chaînes de caractères). Du point de vue du langage C, un tableau est une zone mémoire que l'on peut diviser en n cases élémentaires, n étant la taille du tableau. La taille du tableau est fixée et figée à sa déclaration. Chaque case élémentaire est de la taille du type des éléments du tableau.

Exercice 1:

Écrivez un programme où vous définissez les tableaux suivants et affichez l'espace qu'ils occupent en mémoire.

- un tableau `t1` de 3 entiers
 - un tableau `t2` de 10 caractères
 - un tableau `t3` de 4 entiers non signés initialisé à 0,1,2,3
 - un tableau `t4` de 3 tableaux de 4 flottants
 - un tableau `t5` de 2 tableaux de 2 entiers initialisé comme la matrice identité 2×2
- Affichez aussi la taille de `t4[0]` et `t5[0]`. Qu'en déduisez-vous ?

Modifiez la taille de `t3` pour qu'il comporte 32 éléments, sans en changer l'initialisation. Compilez puis lancez à nouveau votre programme. Que constatez-vous ?

Exercice 2:

Écrivez un programme qui déclare un tableau `t` de 10 entiers initialisé à 0,1,2,3,4,5,6,7,8,9 et qui affiche le contenu de `t` (indiciation : pensez à utiliser vos acquis sur le traitement des files)

Exercice 3:

Écrivez un programme qui remplit un tableau `t` de taille 1000 avec les 1000 premiers entiers, puis qui affiche `t`.

Exercice 4:

Écrivez un programme qui remplit `t` de 11 entiers puis en inverse les éléments. Votre programme affichera `t` avant puis après l'inversion.

Exercice 5:

Écrivez un programme qui déclare deux matrices (tableaux bidimensionnels) de flottants `m1` et `m2` de même dimension, qui en calcule la somme dans une matrice `m3` puis qui affiche cette dernière.

2 Chaînes de caractères

En langage C, les chaînes de caractères sont des tableaux de `char` terminés par la butée `'\0'`. Le caractère `'\0'` n'appartient pas à la chaîne. Il s'agit donc d'une file dont la butée n'est pas traitée comme les autres éléments. La structure de contrôle adaptée au traitement des chaînes de caractères est donc le `while`.

Exercice 6:

Écrivez un programme qui définit une chaîne de caractères, l'initialise à "Bonjour !" puis l'affiche de deux manières différentes.

Exercice 7:

Écrivez un programme qui définit une chaîne de caractères, l'initialise à la valeur de votre choix, en calcule la longueur, et finalement affiche celle-ci.

Exercice 8:

Écrivez un programme qui définit et initialise deux chaînes de caractères et qui les affiche dans l'ordre lexicographique (déterminé lors l'exécution du programme).

Exercice 9:

Écrivez un programme qui définit et initialise une chaîne de caractères et qui la passe en majuscules. Votre programme affichera la chaîne résultante une fois effectuée la transformation

Exercice 10:

Écrivez un programme qui définit deux chaînes `s1` et `s2`, initialise `s1`, la recopie dans `s2` et affiche `s2`. Quel problème peut survenir ? Comment y remédier ?

Exercice 11:

Écrivez un programme qui définit et initialise deux chaînes de caractères `s1` et `s2` aux valeurs de votre choix, puis qui concatène `s2` à `s1`, et enfin qui affiche `s1`. Quel problème peut survenir ? Comment y remédier ?