

Chaînes de caractères - Tableaux et pointeurs

Exercice 1: Fonction d'échange

Écrivez une fonction `echange` qui permet d'échanger deux variables de type `int`. Comment s'écrit l'appel de fonction permettant l'échange de deux variables `a` et `b` ? L'échange des éléments d'indices 3 et 6 d'un tableau `t` (plusieurs solutions) ?

Exercice 2: Longueur d'une chaîne de caractères

Écrire une fonction `strlen`, qui renvoie la longueur (définie comme étant le nombre de caractères précédant le caractère nul, ce dernier non compris) d'une chaîne de caractères.

Exercice 3: Comparaison de chaînes

Écrire une fonction `strcmp` prenant comme arguments deux chaînes de caractères, et renvoyant un entier positif, nul ou négatif, selon que la première chaîne est, respectivement, plus grande, égale, ou plus petite que la seconde dans l'ordre lexicographique. (L'ordre sous-jacent entre les caractères sera celui du codage ASCII.)

Exercice 4: Modification de chaînes

Écrire une fonction `strcpy` qui copie une chaîne de caractères à une adresse passée en argument, et une fonction `strcap` qui transforme une chaîne pour la mettre entièrement en majuscules. La conversion de chaque caractère sera faite au moyen de `toupper`.

Exercice 5: Limitation de taille

La plupart des fonctions écrites lors des exercices précédents, sont dangereuses si elles sont utilisées sans précautions. Expliquer pourquoi.

Écrire une variante de chacune de ces fonctions, avec un paramètre supplémentaire limitant le nombre de caractères examinés.

Note : les fonctions proposées précédemment (sauf `strcap`) sont disponibles dans les bibliothèques standards. Leurs déclarations se trouvent dans le fichier d'en-tête `string.h`; consulter la page de manuel `string` pour plus de détails.

Exercice 6: Duplication de chaînes

Écrire une fonction `strduplicate`, qui réalise et retourne une copie d'une chaîne de caractères. Votre fonction pourra utiliser un tableau de 1000 (par exemple) caractères, déclarée comme une variable globale, et retournera le pointeur `NULL` en cas de manque de mémoire.