



Licence d'informatique
Module de Pratique du C

Travaux dirigés

Manipulation de tableaux

Philippe MARQUET

Octobre 2004

Les exercices proposés illustrent l'utilisation simple de tableaux. En particulier, les chaînes de caractères peuvent être vues comme des tableaux.

Nous reviendrons sur la représentation mémoire des tableaux et leur lien avec les pointeurs dans un prochain TD.

Exercice 1 (Tableau : initialisation et itération)

Quel résultat fournit le programme suivant :

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

static int te[5];
static int tf[5] = {63, 64};

int
main ()
{
    int ta[5] = {17, 15, 13, 12, 14};
    int tb[] = {27, 25, 23, 22, 24};
    int tc[5] = {37, 39};
    int td[5];

    int i;

    printf("ta : ");
    for (i=0 ; i<5 ; i++)
        printf("%d, ", ta[i]);

    printf("\ntb : ");
    for (i=0 ; i<5 ; i++)
        printf("%d, ", tb[i]);

    printf("\ntc : ");
    for (i=0 ; i<5 ; i++)
        printf("%d, ", tc[i]);

    printf("\ntd : ");
    for (i=0 ; i<5 ; i++)
        printf("%d, ", td[i]);

    printf("\nte : ");
    for (i=0 ; i<5 ; i++)
        printf("%d, ", te[i]);

    printf("\ntf : ");
    for (i=0 ; i<5 ; i++)
        printf("%d, ", tf[i]);

    putchar('\n');

    exit (EXIT_SUCCESS);
}
```

Exercice 2 (Copie de tableaux)

Soient les deux tableaux définis par

```
#define SIZE 12
int tsrc[SIZE], tdest[SIZE];
```

Question 2.1 Donnez le code C permettant d'affecter à `tdest` les éléments de `tsrc`.

Question 2.2 Donnez le code C permettant d'affecter à `tdest` les seuls éléments strictement positifs de `tsrc`. On complètera la fin du tableau par des valeurs nulles.

Exercice 3 (Compteur de chiffres et de blancs)

On veut, au sein d'une fonction, compter les occurrences de chaque chiffre ('0' à '9') et des blancs (' ' ou \t) d'un texte lu sur `stdin`. Le résultat sera stocké dans un tableau de 10 entiers pour les chiffres et dans une variable entière pour les blancs.

Exercice 4 (Valeur maximale)

Question 4.1 Donnez la déclaration d'une fonction `max_val()` qui retourne la plus grande valeur, en valeur absolue, d'un tableau de valeurs `double` passé en paramètre.

Question 4.2 Donnez la définition de la fonction `max_val()`.

Question 4.3 Soient les déclarations

```
#define SIZE                200
double m, t[SIZE];
```

donnez le code C qui range dans `m` la plus grande valeur absolue des éléments de `t`.

Exercice 5 (Recherche par dichotomie)

On écrit une fonction qui recherche l'indice d'un nombre `x` dans un tableau `v` de `n` flottants triés par ordre croissant :

```
int search(float x, float v[], int n);
```

Exercice 6 (Crible d'Ératostène)

Le crible d'Ératostène permet d'identifier les nombres premiers inférieurs à un entier donné. Il est basé sur la construction d'une liste contenant initialement tous les entiers. On supprime successivement les entiers multiples du plus petit élément de la liste identifié comme premier.

Proposez une fonction qui affiche les nombres premiers inférieurs à une valeur `N` donnée.

Exercice 7 (Inversion d'une chaîne de caractères)

En langage C, les chaînes de caractères sont rangées sous forme de tableaux de caractères, terminés par le caractère '\0'.

On veut inverser une chaîne de caractères, sans déclarer un second tableau. Pour ce faire, on pourra avoir besoin de la fonction `strlen(const char s[])` déclarée dans `string.h` qui retourne la longueur d'une chaîne (non compris le caractère '\0').

```
void str_reverse(char str[]);
```

Exercice 8 (Conversion décimal/caractères)

La fonction `itoa(int n, char s[])` retourne dans `s` la chaîne de caractères représentant l'entier signé passé en paramètre. Pour l'écrire, on se méfiera du signe, et du fait que l'on ne veut pas de zéros inutiles. Pour cela, on pourra réutiliser la fonction `str_reverse()`.

Exercice 9 (Chaînes de caractères)

Réécrire des fonctions de la bibliothèque de manipulation des chaînes de caractères. Ces fonctions ne gèrent pas l'allocation mémoire, le programmeur doit s'assurer de la validité avant l'appel à ces fonctions.

```
/* recopie le contenu de src dans dest
   retourne dest */
char *strcpy(char *dest, const char *src);

/* recopie src a la fin de dest (concat)
   retourne dest */
char *strcat(char *dest, const char *src);

/* compare les contenus de s2 et s1;
   renvoie la difference entre les deux premiers caracteres differents */
int strcmp(const char *s1, const char *s2);
```