TD/TP 4 : Tableaux, chaînes de caractères

Gilles Peskine

25 octobre 2002

1 Listing d'un tableau

Écrire une fonction qui prend comme paramètres un pointeur sur un tableau d'entiers et sa longueur et qui affiche chaque élément sur une ligne, précédé par son indice dans le tableau.

2 Recherche d'un élément dans un tableau

Écrire une fonction qui prend comme paramètres un pointeur sur un tableau d'entiers a, sa longueur n et un entier x, et qui renvoie le plus petit indice k tel que $a[k] = x$. Si x n'est pas présent dans le tableau, renvoyer -1.

3 Comptage d'occurences

Écrire une fonction qui prend comme paramètres un pointeur sur un tableau d'entiers, sa longueur et un entier x, et qui renvoie le nombre d'occurences de x dans le tableau, c'est-à-dire le nombre d'éléments du tableau qui sont égaux à x.

4 Somme élément par élément de deux tableaux

Écrire une fonction qui prend comme paramètres trois pointeurs vers des tableaux de même longueur et la longueur en question, et qui place dans chaque case i du troisième tableau la somme des éléments i des deux premiers tableaux.

5 Crible d'Ératosthène

Un nombre premier est un nombre entier qui n'est divisible que par 1 et par lui-même. Par exemple, 2 est premier, 13 est premier, mais $8 = 2 \times 4$ et $15 = 5 \times 3$ ne sont pas premiers. On convient que 1 n'est pas premier.

Le crible d'Ératosthène est une méthode pour trouver les nombres premiers compris entre 2 et un nombre fixé N. On commence par écrire tous les nombres de 2 à N. Ensuite, on raye tous les nombres multiples de 2, à l'exception de 2 lui-même : 2 est premier, et on raye 4, 6, 8, etc ; et 2 est premier. On recommence avec 3 : 3 est premier, et fon raye 9, 15, etc. (6, 12, etc. sont déjà rayés). Puis avec le nombre suivant non rayé (5), et ainsi de suite. On s'arrête quand tous les nombres jusqu'à N sont soit premiers soit rayés.

Écrire un programme qui utilise cette méthode pour trouver et afficher les nombres premiers inférieurs à N (on pourra tester avec $N = 100$ par exemple).

Question subsidiaire (de mathématiques) : en fait, il y a un nombre L tel que dès que tous les nombres inférieurs à L sont rayés ou premiers, les nombres non rayés entre L et N sont tous premiers. Que vaut L en fonction de N ?

6 Chaînes de caractères

Une chaîne de caractères est stockée sous la forme d'un tableau de caractères. La fin de la chaîne est marquée par le caractère nul (qu'on peut écrire `0` ou `'\000'`, à ne pas confondre avec le caractère `'0'` qui représente un chiffre). Les caractères du tableau au-delà du premier caractère nul sont ignorés.[1]

(a) Écrire une fonction qui calcule la longueur d'une chaîne de caractères. Par exemple : `longueur("bon jour")` $\Longrightarrow$ `8`.

(b) Écrire une fonction qui renvoie l'indice de la première occurence d'un caractère dans une chaîne. La fonction renverra -1 si le caractère n'est pas présent dans la chaîne.

(c) Même question, mais en cherchant la dernière occurence plutôt que la première.

7 Comparaison de deux chaînes

(a) Si `a` et `b` sont deux chaînes de caractères, que signifie l'expression `(a == b)` ?

(b) Écrire une fonction qui compare deux chaînes de caractères. La fonction devra renvoyer 1 si les chaînes sont identiques et 0 sinon. Par exemples :

`chaines_identiques("coucou", "coucou")` $\Longrightarrow$ `1`
`chaines_identiques("coucou", "bonjour")` $\Longrightarrow$ `0`
`chaines_identiques("cou", "coucou")` $\Longrightarrow$ `0`

8 Écriture décimale

(a) Écrire une fonction qui prend comme paramètre une chaîne de caractères représentant un nombre entier sous forme décimale et qui renvoie le nombre entier correspondant. On suppose que la chaîne ne contient que des chiffres. Par exemple : `decimal("42")` $\Longrightarrow$ `42`.

(a') Même question, en supposant que la chaîne peut optionnellement commencer par un signe `+` ou `-`.

(b) Écrire une fonction qui prend comme paramètre un nombre entier et qui écrit son écriture décimale dans une chaîne allouée globalement.

```
char tampon[50];
void ecriture_decimale(unsigned int x) { ... }
```

(c) Par quoi faut-il remplacer le nombre 50 pour être sûr d'avoir un tableau de caractères de longueur suffisante ? On utilisera les faits suivants :

- L'opérateur `sizeof` permet de connaître le nombre de *bytes* qu'occupe une variable d'un type donné. Par exemple, `sizeof(unsigned int)` est une constante entière, dont la valeur est le nombre de *bytes* occupés par un `unsigned int`.
- La constante `CHAR_BIT`, définie dans l'en-tête `limits.h`, indique le nombre de bits (chiffres binaires) dans un *byte*. On pourra utiliser l'approximation suivante : un chiffre décimal suffit pour représenter trois bits.

(c') Autre méthode : utiliser la constante `UINT_MAX` définie dans `limits.h`, qui est le plus grand nombre représentable dans une variable de type `unsigned int`.

[1] On remarque que ceci implique qu'une chaîne ne peut pas contenir de caractère nul.

9 Histogrammes

On se donne un tableau d'entiers positifs `a` à `n` éléments. Un histogramme correspondant à ce tableau est un graphique qui contient pour chaque entrée du tableau un rectangle dont la longueur est proportionnelle à la valeur de l'entrée du tableau.

(a) Utiliser la bibliothèque `graphics` pour écrire une fonction qui dessine l'histogramme correspondant à un tableau.

(b) Utiliser la fonction `putchar` pour afficher un histogramme en mode texte, comme ceci :

```
******
**

***
```

(pour le tableau `{6, 2, 0, 3}`).

(c) Même question, mais on veut dessiner l'histogramme dans l'autre sens :

```
*
*
*
*  *
** *
** *
```

10 Ligne de commande

Il existe deux manières de déclarer la fonction `main` :

- `int main ()`, ce qui est équivalent à `int main (void)`.
- `int main (int argc, char * * argv)`. Dans ce cas, `argv` est un pointeur sur un tableau de chaînes de caractères. `argv[0]` est le nom du programme, et les éléments suivants `argv[1]`, `argv[2]`, etc., sont les arguments passés sur la ligne de commande du programme, jusqu'à `argv[argc-1]`. Enfin `argv[argc]` est un pointeur nul (`NULL`).

(a) Écrire un programme qui affiche les arguments qu'on lui passe, un par ligne. Par exemple :

```
[toto@lenteja tp04]$ ./affiche_arguments bonjour tout le monde
bonjour
tout
le
monde
```

(b) Écrire un programme qui suppose que tous ses arguments sont des nombres entiers écrits sous forme décimale et qui calcule et affiche leur somme. Par exemple :

```
[toto@lenteja tp04]$ ./somme 3 42 25 18
88
```

(c) L'en-tête `stdlib.h` définit deux valeurs entières `EXIT_SUCCESS` et `EXIT_FAILURE` ; par convention, pour indiquer qu'un programme échoue, on lui fait renvoyer `EXIT_FAILURE`, tandis qu'on lui fait renvoyer 0 ou `EXIT_SUCCESS` s'il réussit. Adapter le programme du *(b)* pour qu'il détecte si un de ses arguments n'est pas un nombre entier. Dans ce cas, ne rien afficher, mais faire retourner au programme la valeur `EXIT_FAILURE`.

11 Recherche d'une sous-chaîne

Écrire une fonction qui recherche une chaîne dans une autre. Renvoyer le plus petit indice auquel la sous-chaîne commence, ou -1 si la sous-chaîne n'est pas présente. Exemples :

`recherche_sous_chaine("bonjour", "jour")` $\Longrightarrow 3$
`recherche_sous_chaine("coucou", "ou")` $\Longrightarrow 1$
`recherche_sous_chaine("bonjour", "bnjr")` $\Longrightarrow -1$

12 Recherche d'un élément dans un tableau trié

On veut écrire une fonction qui recherche un élément x dans un tableau d'entiers triés en ordre croissant. Bien sûr, on peut utiliser la fonction de l'exercice 2, mais il y a plus efficace.

(a) On peut interrompre la recherche dès qu'on tombe sur un élément trop grand. Faire cette modification.

(b) Recherche binaire. Au lieu de regarder les éléments du tableau de gauche à droite, on commence par regarder l'élément du milieu a. Si a est égal à x, on a gagné. Si a est plus grand que x, on sait que x ne peut être que avant a : on a éliminé la moitié de l'espace de recherche en une comparaison. De même, si $a < x$, on n'a plus qu'à chercher x dans la deuxième moitié du tableau. Tant qu'on n'a pas trouvé x, on applique cette méthode à un sous-tableau dont la taille diminue de moitié à chaque coup. Si on arrive à un limiter x à un sous-tableau de taille 0, on sait que x est absent.