

INFO-H-100 - Programmation

TP 12 - Exercices de synthèse

Juin 2003 – Question 2

Soit un vecteur V de taille n contenant des caractères. Ces caractères peuvent être les 26 lettres majuscules de l'alphabet ainsi que les symboles $+$, $-$, $*$ et $/$.

On désire trier le vecteur V mais non pas suivant l'ordre alphabétique "classique" mais bien suivant un ordre défini par le contenu d'un autre vecteur. Ce deuxième vecteur, appelons le O , est de taille 30 et il contient une seule occurrence de chacun des caractères cités ci-dessus. Ce vecteur indique l'ordre *croissant* des caractères utilisés pour trier les éléments du vecteur v .

Exemple :

V

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | + | F | - | + | K | A | B | G |
|---|---|---|---|---|---|---|---|---|

O

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|-----|
| F | + | K | B | - | A | G | R | P | Z | U | M | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|-----|

V trié

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| F | + | + | K | B | - | A | A | G |
|---|---|---|---|---|---|---|---|---|

On vous demande d'écrire une fonction *tri_alpha* qui recevra deux vecteurs en paramètres (V et O) et qui réalisera le tri décrit ci-dessus. Cette fonction de tri devra obligatoirement se baser sur la méthode du **tri bulle**.

Juin 2007 – Question 3

En informatique, il est possible de représenter un polynôme sous la forme d'un vecteur P où chaque $P[i + 1]$ donne le coefficient d'indice i du polynôme. C'est-à-dire

$$P^{(n)}(x) = \sum_{i=0}^n a_i x^i \iff P[i + 1] = a_i$$

et le premier élément $P[0]$ contient le degré n du polynôme que représente le vecteur P .

On décide de stocker plusieurs polynômes dans un tableau à deux dimensions de telle sorte que chaque ligne du tableau représente un polynôme donné selon l'encodage fourni ci-dessus.

Lorsqu'on souhaite comparer des polynômes selon leur valeur respectivement obtenue pour un réel x donné très grand, une bonne première approximation consiste à comparer les polynômes selon leur degré et en cas d'égalité selon leur coefficient de ce degré :

$$P_1^{(m)} > P_2^{(n)} \iff (m > n) \vee ((m == n) \wedge (a_m > b_n))$$

où $a_i = P_1[i + 1]$ et $b_j = P_2[j + 1]$. On demande d'écrire une fonction **triPoly** qui :

- reçoit un tableau d’entiers de dimension $MAX \times MAX$ qui contient les polynômes comme décrit ci-dessus (c-à-d le degré du polynôme concerné et ses coefficients),
- et trie l’ensemble des polynômes en utilisant un tri par insertion et la relation d’ordre fournie.

Aout 2007 – Question 3

L’informatique a souvent permis de remplacer un système papier. C’est par exemple le cas du carnet d’adresse. Afin que ce dernier soit efficace, la première chose à faire est de s’assurer d’un bon tri lexicographique ¹ suivant le prénom et le nom des personnes enregistrées.

Le carnet d’adresse sera un tableau de chaînes de caractères. Le nombre d’entrées ne pourra pas dépasser une constante `int MAX` et le tableau sera accompagné d’un paramètre `n` donnant le nombre d’entrées enregistrées. La longueur maximale de chaque entrée sera fixée par la constante `int LMAX` et on utilisera la convention du caractère `'\0'` pour signaler la fin de la chaîne de caractères. Pour des raisons de facilité, **on définira un type `entree` comme étant un tableau de caractères** de capacité maximale `LMAX` et **un type `carnet` comme étant un tableau d’entrees** de capacité maximale `MAX`.

On vous demande d’écrire au moins trois fonctions :

1. la fonction `void triCarnet(carnet C, int n)` qui triera les `n` premières entrées du carnet `C` selon un tri par insertion,
2. la fonction `int compareEntrees(entree e1, entree e2)` qui comparera deux `entrees` `e1` et `e2` selon l’ordre lexicographique et qui renverra une valeur entière négative si `e1` est après `e2`, positive dans le cas inverse et nulle si les deux `entrees` sont strictement identiques,
3. et la fonction `void echangeEntrees(carnet C, int l1, int l2)` qui échangera les deux `entrees` dans le carnet `C` aux index `l1` et `l2`.

Bien entendu, il est également nécessaire de définir les types et constantes utilisés (`MAX`, `LMAX`, `entree` et `carnet`).

¹Les chiffres se trouvent avant les majuscules qui se trouvent avant les minuscules. On ne considérera pas les caractères spéciaux autres qu’alphanumériques.