

Pointeurs, Fonctions, Boucles et Tableaux : exercices

Antoine FRÉNOY,
antoine.frenoy@inserm.fr

13 et 20 novembre 2013

Préliminaire : saisie de tableaux

- Écrivez une fonction `enttab` qui permet à l'utilisateur de rentrer au clavier un tableau d'entiers positifs ou nuls. Cette fonction ne crée pas le tableau (elle prend en arguments l'adresse de départ du tableau déjà existant et son nombre d'éléments) mais le remplit.
- De même écrivez une fonction `entstr` qui permet la saisie d'une chaîne de caractères (c'est à dire un tableau de caractères finissant par le caractère nul, code ASCII 0).

Vous pourrez utiliser ces deux fonctions pour tester vos programmes dans les exercices suivants.

Exercice 1 : Tri de tableaux

1. On veut écrire une fonction qui prend en arguments un pointeur vers entier et un entier, représentant l'adresse de départ et le nombre d'éléments d'un tableau d'entiers, et qui donne en sortie le plus petit élément de ce tableau. Écrivez deux versions de cette fonction : la première, `mintab1`, renvoie un entier, la seconde, `mintab2`, utilise un pointeur vers un argument supplémentaire ("paramètre en sortie" dans le vocabulaire du cours).

Pour les questions suivantes, vous êtes libres de choisir comment vous donnerez le résultat d'une fonction.

2. Écrivez une fonction `mintabind` qui prend en arguments un pointeur vers entier et un entier, représentant l'adresse de départ et le nombre d'éléments d'un tableau d'entiers, et dont le résultat est l'index du plus petit élément de ce tableau.
3. Écrivez une fonction `tabdel` qui prend en arguments un pointeur vers entier et un entier, représentant l'adresse de départ et le nombre d'éléments d'un tableau d'entiers `t`, et un entier `i`, et supprime l'élément à la position `i` du tableau `t`.
4. Écrivez une fonction `inttabsort` qui prend en arguments un pointeur vers entier `p1` et un entier `n`, représentant l'adresse de départ et le nombre d'éléments d'un tableau d'entier `t`, et qui trie ce tableau : après l'exécution `p1` pointe vers l'adresse de départ d'un tableau qui contient les mêmes éléments que `t` mais triés par ordre croissant.

Exercice 2 : Récursivité

1. Soit la fonction $A : \mathbb{N} * \mathbb{N} \rightarrow \mathbb{N}$ définie par :
 - $A(0, n) = n + 1$
 - $A(m, 0) = A(m - 1, 1)$ si $m > 0$
 - $A(m, n) = A(m - 1, A(m, n - 1))$ si $m > 0$ et $n > 0$Écrivez une fonction prenant en arguments deux entiers `m` et `n` et renvoyant $A(m, n)$.
2. Soit $x \in \mathbf{R}$ et $n \in \mathbf{N}$. On a la relation suivante :
 - Si n est pair alors $x^n = (x^{\frac{n}{2}})^2$
 - Sinon $x^n = x * (x^{\frac{n-1}{2}})^2$Écrivez une fonction `exporapide` prenant deux arguments `x` et `n` et renvoyant x exposant `n` en utilisant la méthode décrite ci-dessus.

Exercice 3 : Autres manipulations simples de tableaux

1. Écrivez une fonction `remchar` prenant en arguments une chaîne de caractères (*ie* un tableau de caractères dont la fin est signalée par un caractère de code ASCII 0) et un caractère `c` et supprimant toutes les occurrences du caractère `c` dans la chaîne.
2. Écrivez une fonction `ispal` prenant en argument un tableau d'entiers et indiquant s'il s'agit d'un palindrome (un palindrome est un tableau qui ne change pas lorsqu'il est lu à l'envers, par exemple `[1, 5, 3, 5, 1]`).
3. Écrivez une fonction `freqchar` prenant en argument une chaîne de caractères et retournant le caractère le plus utilisé dans cette chaîne.

Exercice 4 : Polynômes

On représente un polynôme par un entier (son degré appelé par exemple `d`) et un tableau (appelé par exemple `t`) de flottants tel que $\forall n \leq d, t[n]$ soit le coefficient de degré n .

1. Écrivez une fonction `evalpol` prenant en arguments un entier et un tableau de flottants représentant un polynôme ainsi qu'un flottant `x` et renvoyant la valeur du polynôme en `x`.
2. Écrivez une fonction `addpol` prenant en arguments deux polynômes (c'est à dire deux entiers et deux tableaux de flottants) et donnant en sortie la somme des deux polynômes (c'est à dire un entier et un tableau de flottants représentant le polynôme somme).
3. Écrivez une fonction `mulpol` prenant en arguments deux polynômes et donnant en sortie le produit des deux polynômes.
4. Écrivez une fonction `derpol` prenant en argument un polynôme et donnant en sortie sa dérivée.
5. Écrivez une fonction `zerpol` prenant en arguments un polynôme `Q` de degré impair et de coefficient de plus haut degré strictement positif, et un flottant `p`, et retournant un zéro du polynôme avec une précision `p`, *ie* retournant un flottant `x` tel que $|Q(x)| < p$.

Exercice 5 : Quelques questions plus avancées

1. Écrivez une fonction `subtab` qui prend en arguments deux chaînes de caractères `s1` et `s2` et qui indique si `s1` est une sous-chaîne de `s2` (c'est à dire si on peut obtenir `s1` en supprimant des lettres de `s2`).
2. La distance de Levenshtein indique, pour deux chaînes de caractères (de longueurs éventuellement différentes) le nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre. Écrivez une fonction `levenshtein` prenant en arguments deux chaînes de caractères et renvoyant leur distance de Levenshtein.