ÉCOLE POLYTECHNIQUE

CONCOURS D'ADMISSION 2003 FILIÈRES **PSI** ET **PT**

ÉPREUVE FACULTATIVE D'INFORMATIQUE

(Durée : 2 heures)

L'utilisation des calculatrices **n'est pas autorisée** pour cette épreuve.

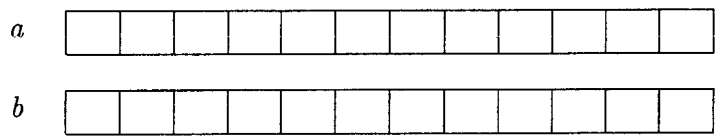
⋆⋆⋆

Avertissements :
Le langage de programmation choisi par le candidat doit être spécifié en tête de la copie.
On attachera une grande importance à la concision, à la clarté, et à la précision de la rédaction.

Le plan de la rue

Après un grand chamboulement, une administration défaillante tente de reconstituer l'ordre de voisinage dans une rue de n maisons ($n > 0$). Pour simplifier, on suppose la rue occupée sur un seul de ses deux cotés. Chaque propriétaire a un identifiant qui est un nombre entier le désignant sans ambiguïté. Chaque maison a deux maisons voisines (à sa gauche et à sa droite). On demande à chaque propriétaire de maison d'écrire son identifiant a_i et l'identifiant de son voisin de droite b_i sur une fiche ($0 \leq i < n$). Par convention, l'identifiant du voisin de droite de la maison située à l'extrémité droite de la rue est -1. On collecte les n fiches et on veut reconstituer le plan de la rue dans l'ordre initial.

On représente les n fiches par deux tableaux globaux a et b de longueur n, contenant des entiers naturels a_i et b_i ($0 \leq i < n$), sauf pour une valeur k de l'indice où on a $b_k = -1$.



On suppose, dans les deux premières parties (questions 1 à 4), que chaque propriétaire ne possède qu'une seule maison. Donc, tout entier naturel apparaît au plus une fois dans chaque tableau.

Le temps d'exécution $T(f)$ d'une fonction f est le nombre d'opérations élémentaires (addition, soustraction, multiplication, division, affectation, etc.) nécessaire au calcul de f. Lorsque ce temps d'exécution dépend d'un paramètre n, il sera noté $T_n(f)$. On dit que la fonction f s'exécute :

- en temps linéraire en n, s'il existe $K > 0$ tel que pour tout n, $T_n(f) \leq Kn$;
- en temps quadratique en n, s'il existe $K > 0$ tel que pour tout n, $T_n(f) \leq Kn^2$.

I – Reconstitution de la rue

Question 1 Écrire une fonction plusadroite() qui retourne, en temps linéaire en n, l'indice k (dans le tableau a) de la maison la plus à droite dans la rue.

Question 2 Écrire une fonction calculerG() qui calcule, en temps quadratique en n, les valeurs du tableau g tel que $g[i]$ est l'indice du voisin de gauche de la maison d'indice i $(0 \leq i < n)$. (On supposera que g est un tableau global de n éléments; et on posera $g[\ell] = -1$ quand ℓ est l'indice de la maison la plus à gauche dans la rue).

Question 3 En supposant le tableau global g donné par la question précédente, en déduire la fonction imprimer() qui imprime en temps linéaire en n tous les identifiants de propriétaires des maisons dans l'ordre initial de la rue de la gauche vers la droite. (On construira un tableau intermédiaire où on rangera les propriétaires dans l'ordre de la rue).

II – Optimisation de la solution

On suppose à présent les tableaux a et b agencés de telle sorte que le tableau b est trié dans l'ordre croissant. On a donc toujours $b_i < b_j$ pour $0 \leq i < j < n$.

La fonction calculerG() peut maintenant utiliser une recherche dichotomique pour déterminer l'indice du voisin de gauche de toute maison i. Pour trouver l'indice j tel que $b_j = a_i$, on compare la valeur a_i à la valeur b_m du milieu du tableau b et on recommence, si nécessaire, sur la moitié haute ou basse du tableau en fonction de l'ordre relatif de b_m et a_i.

Question 4 Écrire une nouvelle version calculerG1() de la fonction calculerG() utilisant la recherche dichotomique. Donner un ordre de grandeur de son temps d'exécution en fonction de n.

III – Propriétés multiples

On se remet à nouveau dans les conditions de la première partie, en ne faisant aucune supposition sur l'ordre du tableau b.

Un propriétaire peut maintenant avoir plusieurs maisons dans la rue et écrit autant de fiches, donnant son identifiant et celui de son voisin de droite, qu'il a de maisons. Chaque propriétaire a_i connait l'ordre d'apparition de ses maisons dans la rue. Pour chaque maison i $(0 \leq i < n)$, un tableau global h donne l'indice de la maison suivante à gauche dans la rue appartenant à a_i si elle existe. Sinon $h[i] = -1$. On a donc $a[h[i]] = a[i]$ quand $h[i] > 0$.

À partir des tableaux a, b et h, l'Administration centrale a déjà pré-calculé le tableau global g donnant l'indice du voisin de gauche dans la rue. Dans le cas où le voisin de gauche possède plusieurs maisons, la valeur indiquée par $g[i]$ est toujours l'indice de la maison la plus à droite dans la rue appartenant à $a[g[i]]$. Remarque : la maison indiquée par $g[i]$ n'est pas forcément à la gauche de la maison i. Dans le cas où i est l'indice de la première maison à gauche, $g[i]$ vaut -1.

On dispose donc des quatre tableaux globaux suivants :

a												
b												
g												
h												

Pour retrouver l'ordre initial de la rue, on procède comme suit. On démarre avec la maison la plus à droite dans la rue en se servant du tableau b. On avance dans le tableau a grâce au tableau g. Si on rencontre un propriétaire qui a plusieurs maisons, on continue avec un des voisins de gauche possibles dans la rue. L'ordre donné par le tableau h permet de trouver le bon voisin de gauche, et on continue.

Question 5 Reconstruire l'ordre de la rue dans le cas où on a $n = 7$ et les tableaux a, b, g, h tels que $a = \langle 1, 2, 8, 2, 1, 1, 7 \rangle$, $b = \langle 2, 1, 2, -1, 7, 8, 1 \rangle$, $g = \langle -1, 5, 5, 2, 3, 6, 5 \rangle$ et $h = \langle -1, -1, -1, 1, 0, 4, -1 \rangle$.

Question 6 En déduire la fonction imprimer2() qui imprime, en temps quadratique en n, tous les identifiants des propriétaires des maisons dans l'ordre initial de la rue de la gauche vers la droite.

Question 7 Écrire une fonction imprimer3() qui fait la même impression en temps linéaire en n.

* *
*