

Devoir surveillé

25 octobre 2018 — Durée 1h15

Document autorisé : **Mémento C** vierge de toute annotation manuscrite

On considère le paquetage **recherche**, qui fournit une fonction `recherche_indice`. Cette fonction prend en argument un entier `x` et une séquence `t` (de type `tableau_entiers`) *triée par ordre croissant*, et renvoie l'indice de la première occurrence de `x` dans `t`. La fonction renvoie la valeur `-1` si `t` ne contient pas la valeur `x`.

L'implémentation de ce paquetage est donnée ci-dessous :

```
1 #include "recherche.h"
2
3 /* Données : un entier x, un tableau d'entiers trié t
4  Renvoie :
5   - l'indice dans le tableau de la première occurrence de x
6   - -1 si le tableau ne contient pas x
7  Précondition : le tableau t est trié par ordre croissant
8  */
9 int recherche_indice(int x, tableau_entiers t) {
10     int ib, ih, im;
11
12     /* Recherche dichotomique */
13     ib = 1;
14     ih = t.taille-1;
15     while (ib < ih) {
16         im = (ib + ih) / 2;
17         if (t.tab[im] < x) {
18             ib = im+1;
19         } else {
20             ih = im;
21         }
22     }
23     if (t.tab[ib] == x) {
24         return ib;
25     } else {
26         return -1;
27     }
28 }
```

Les paquetages `type_tableau` et `es_tableau` sont donnés en annexe.

Exercice 1. (2 pt)

Écrire le fichier *spécification* du paquetage **recherche**.

Exercice 2. (4 pt)

Écrire un programme de test du packaging **recherche**. Ce programme doit utiliser les packages **type_tableau** et **es_tableau** fournis en annexe, pour lire un tableau dans un fichier dont le nom est donné en argument de la ligne de commande, lire ensuite un entier **x** dans ce même fichier, et afficher le résultat de l'appel de la fonction **recherche_indice**.

Exercice 3. (5 pt)

Décrire un jeu de tests fonctionnels permettant de tester la fonction **recherche_indice** à l'aide du programme écrit à l'exercice précédent.

NB : il est demandé de *décrire* ce jeu de tests, et donc de *justifier* sa construction, sans nécessairement écrire explicitement les valeurs de chaque test.

Exercice 4. (2 pt)

La fonction **recherche_indice** comporte au moins une erreur.

Identifier une de ces erreurs. Donner un test permettant de mettre cette erreur en évidence.

NB : il n'est pas demandé de *corriger* l'erreur trouvée!

Exercice 5. (4 pt)

On souhaite maintenant utiliser un *oracle* pour vérifier automatiquement que le résultat fourni par la fonction **recherche_indice** est correct. Quelles sont les propriétés à vérifier par cet oracle? Écrire une fonction oracle vérifiant ces propriétés.

Exercice 6. (3 pt)

Écrire un Makefile permettant de compiler le programme écrit à l'exercice 2.

Annexes : packages utilisés

Paquetage **type_tableau**

```
1 #ifndef _TYPE_TABLEAU_H_
2 #define _TYPE_TABLEAU_H_
3
4 #define TAILLE_MAX 10000
5
6 /* Définition du type vecteur_entiers :
7    tableau d'entiers de taille TAILLE_MAX */
8 typedef int vecteur_entiers[TAILLE_MAX];
9
10 /* Structure contenant un tableau (de taille TAILLE_MAX) et un entier
11    taille : le nombre d'entiers du tableau */
12 typedef struct {
13     int taille;
14     vecteur_entiers tab;
15 } tableau_entiers;
16
17 #endif /* _TYPE_TABLEAU_H_ */
```

Paquetage es_tableau

```
1 #ifndef _ES_TABLEAU_H_
2 #define _ES_TABLEAU_H_
3
4 #include <stdio.h>
5 #include "type_tableau.h"
6
7 void lire_tableau(FILE * fichier, tableau_entiers * t);
8
9 void ecrire_tableau(FILE * fichier, tableau_entiers t);
10
11 #endif /* _ES_TABLEAU_H_ */
```

```
1 #include <stdio.h>
2 #include "es_tableau.h"
3
4 void lire_tableau(FILE * fichier, tableau_entiers * t) {
5     int i;
6
7     /* Lecture de la taille du tableau */
8     fscanf(fichier, "%d", &(t->taille));
9
10    /* Lecture des valeurs du tableau */
11    for (i = 0; i < t->taille; i++) {
12        fscanf(fichier, "%d", &(t->tab[i]));
13    }
14 }
15
16 void ecrire_tableau(FILE * fichier, tableau_entiers t) {
17     int i;
18
19     /* Écrire la taille du tableau dans le fichier */
20     fprintf(fichier, "%d\n", t.taille);
21
22     /* Écrire les valeurs du tableau */
23     for (i = 0; i < t.taille; i++) {
24         fprintf(fichier, "%d\n", t.tab[i]);
25     }
26 }
```