

TP 8 Partie 1

Les tableaux à 1 dimension : algorithmes de recherche

Exercice 1 : la recherche séquentielle

Soit le type `tabEntiers` défini comme suit :

```
#define N 10  
typedef int tabEntiers[N];
```

Question 1

Écrire et tester une fonction `int recherche_sequentielle (int valeur, tabEntiers tablo)`, qui recherche de manière **séquentielle** si `valeur` est présent dans le tableau `tablo` ; si oui, la fonction délivre l'indice auquel se trouve `valeur` dans `tablo`, et délivre -1 sinon.

N.B. Quand on commence la recherche, on ne sait pas combien de fois on va effectuer une comparaison, donc il n'est pas question d'utiliser une boucle *pour* : utilisez une boucle *"tant que"* à deux conditions.

Question 2

Une variante de cet algorithme consiste à utiliser une sentinelle. Cela suppose de modifier le type `tabEntiers`:

```
typedef int tabEntiers[N+1] ;
```

l'élément d'indice `N` servant à contenir la sentinelle.

Pour le test, on pourrait prendre comme tableau effectif un tableau initialisé par exemple comme suit :

```
tabEntiers leTablo = {23, 54, 65, 12, 43, 78, 68, 93, 18, 31} ;
```

Exercice 2 : la recherche dichotomique

Soit le type `tabEntiers` défini comme suit :

```
#define N 10
typedef int tabEntiers[N] ;
```

Question 1 : méthode itérative

Écrire et tester une fonction :

```
int recherche_dicho_iteratif(int valeur, tabEntiers tablo)
```

qui recherche par **dichotomie** si *valeur* est présent dans *tablo*, sachant que *tablo* est trié ; si oui la fonction délivre l'indice de *valeur* dans *tablo* , -1 sinon.

Cette recherche utilise le raisonnement suivant.

Notons $t[i..j]$ le sous-tableau du tableau t contenant les éléments d'indice i à j ; l'élément recherché est comparé avec l'élément milieu du sous-tableau dans lequel on raisonne. Trois cas sont à envisager :

- La valeur recherchée est égale à l'élément milieu : on a donc trouvé et la recherche est terminée.
- La valeur recherchée est supérieure à l'élément milieu : la recherche continue dans la moitié « supérieure » du sous-tableau.
- La valeur recherchée est inférieure à l'élément milieu : la recherche continue dans la moitié « inférieure » du sous-tableau.

Si on aboutit à une recherche dans un sous-tableau $t[i..j]$ tel que j est inférieur à i , alors c'est que la valeur recherchée n'est pas présente dans le tableau t .

Pour le test, on pourrait prendre comme tableau effectif un tableau initialisé comme suit, par exemple :

```
tabEntiers leTabloTrie = {13, 24, 35, 42, 53, 68, 77, 83, 88, 91} ;
```

Question 2 : méthode récursive

Écrire et tester une fonction récursive

```
int recherche_dicho_recurusif(int valeur, tabEntiers tablo,
                              int debut, int fin)
```

qui recherche par dichotomie si *valeur* est présent dans le sous-tableau $t[debut..fin]$, sachant que t est trié ; si oui, la fonction délivre l'indice de *valeur* dans *tablo* , -1 sinon.

Exercice 3

Dans cet exercice, on se propose de traiter des mots de 8 lettres sous la forme de tableaux de 8 caractères. Par exemple, le mot PISTACHE sera représenté par le tableau :

P	I	S	T	A	C	H	E
---	---	---	---	---	---	---	---

Vous définirez en entête de votre programme le type **mot8lettres** de la manière suivante :

```
#define N 8
typedef char mot8lettres[N] ;
```

Question 1

Écrivez une procédure **init** qui possède un paramètre de type **mot8lettres** et qui initialise tous ses éléments avec le caractère ' * '.

Question 2

Écrivez une procédure **saisir** qui fournit en sortie un paramètre de type **mot8lettres**, après avoir lu au clavier une série 8 caractères

Question 3

Écrivez une procédure **afficher** qui affiche à l'écran tous les éléments d'un mot de 8 lettres passé en paramètre.

Question 4

Écrivez une fonction **contient** qui possède en paramètres d'entrée un mot de 8 lettres et un caractère. La fonction doit retourner **true** si le caractère est contenu dans le mot de 8 lettres, et retourner **false** sinon.

Question 5

Écrivez un main pour tester les procédures et fonctions précédentes. Votre main devra :

- initialiser une variable de type **mot8lettres**,
- initialiser ce mot par une saisie clavier,
- afficher le contenu de ce mot,
- indiquer si le caractère 'x' est présent dans ce mot.