

Algorithmique TD2:

Tableaux: recherche et modification d'éléments

16 octobre 2009

1. Écrire une procédure qui inverse les éléments d'un tableau :
 - le premier devient le dernier
 - le deuxième devient l'avant dernier
 - etc.
2. Écrire une procédure qui supprime d'un tableau d'entiers la valeur min et la valeur max.
3. Écrire une fonction qui à partir de deux tableaux de même taille passés en paramètres, construit un nouveau tableau contenant la somme des éléments de mêmes indices.
4. Fusion et suppression des doublons :

Soient deux tableaux d'entiers triés par ordre croissant. Écrire une fonction qui fusionne les deux tableaux en supprimant les doublons. Le tableau final doit bien sûr être également trié.
5. Nombres premiers :

Écrire une fonction qui retourne un tableau contenant l'ensemble des nombres premiers inférieurs à 100. On utilisera le crible d'Ératostène dont le principe est le suivant :

 - on initialise la première case du tableau avec le nombre premier 2.
 - on cherche ensuite un nombre premier supérieur à 2 qui possède la propriété de n'avoir que 2 diviseurs distincts (1 et lui-même) et on l'insère dans le tableau
 - on réitère le traitement pour les 100 premiers nombres
6. Cryptographie :
 - (a) Écrire une fonction permettant de crypter un message selon le code de César dont le principe consiste à décaler chaque lettre du message d'une valeur passée en paramètre. Par exemple, la lettre 'a' décalée de 3 se transforme en la lettre 'd' tout comme la lettre 'w' décalée de 7. On utilisera un tableau pour stocker l'ensemble des lettres de l'alphabet. Pour simplifier le traitement, on supposera que les messages sont des tableaux de caractères.
 - (b) Écrire une nouvelle fonction de cryptage selon le principe suivant :

- additionner chaque lettre du message avec la lettre suivante de l'alphabet
- normaliser la valeur obtenue pour qu'elle corresponde à une lettre de l'alphabet
- utiliser le code de César sur la valeur obtenue

7. Soit la fonction suivante vue en cours :

```

fonction recherche (tableau tab(15):entier , taille:entier ,elem:
entier):booleen
variables
  deb:entier
  fin:entier
  mil:entier
deb ← 1
mil ← taille/2
fin ← taille
tant que deb<fin et tab(mil) ≠ elem faire
  si tab(mil) < elem alors
    deb ← mil+1
  sinon
    fin ← mil-1
  fsi
  mil ← (fin+deb)/2
ftq
retourner tab(mil)=elem
finfonction

```

- (a) Trouver un invariant pour cette boucle
- (b) Quelle est le coût de cette fonction