

TD - Arbres binaires

L'objectif de ce TD est la réalisation d'un programme permettant d'effectuer différentes opérations sur des ensembles d'entiers (un ensemble ne doit pas contenir d'éléments redondants). Il se décompose en deux parties :

1. *Représentation d'un ensemble par un arbre binaire ordonné (Arbre Binaire de Recherche).*

Il s'agit ici de définir le type ensemble permettant de coder un ensemble d'entiers par un arbre binaire ordonné.

2. *Quelques Opérations.*

Ecrire les procédures et fonctions primitives nécessaires à la gestion d'un ensemble.

- cardinal de l'ensemble
- test d'appartenance
- ajout d'un élément
- affichage des éléments de l'ensemble.
- réinitialiser l'ensemble à vide en libérant l'espace mémoire occupé par les éléments de l'ensemble.
- copie d'un ensemble dans un autre
- réunion de deux ensembles
- intersection de deux ensembles
- suppression d'un élément.

Examen - Programmation Avancée
K. Gaber

On souhaite gérer les livres d'une bibliothèque à l'aide d'une table (**Biblio**) de listes chaînées.

A chaque nom d'auteur correspondra un code compris entre 0 et M-1. Chaque code donne accès à une liste chaînée triée de tous les noms d'auteurs ayant le même code.

Pour chaque auteur, on connaît son nom : **nomAuteur** ; on dispose également de deux pointeurs **premier** et **dernier** sur la liste des livres écrits par l'auteur, **triée par ordre chronologique** : **premier** est le pointeur sur le premier livre écrit par l'auteur alors que le **dernier** pointe sur le dernier livre écrit par l'auteur.

On supposera que tous les auteurs ont un nom différent et que les titres de tous les livres sont différents.

On dispose d'une fonction **calculCode** qui calcule le code d'un auteur. L'entête de cette fonction est : **int calculCode(char *nom) ;**

Spécification : retourne le code, compris entre 0 et M-1, associé au nom.

1. Définir les types et structures de données nécessaires.
2. Donner une représentation schématique de la structure de données (utiliser un exemple).
3. Ecrire une fonction **nbAut(Auteur *p)** qui retourne le nombre d'auteurs présent dans la liste des auteurs.
4. Ecrire une fonction **nbAutTot(Biblio t)** qui retourne le nombre d'auteurs présent dans la bibliothèque.
5. Ecrire une fonction qui retourne le nombre de livre écrit par un auteur donné.
6. Ecrire une fonction ou procédure qui insère un auteur dans la bibliothèque.
7. Ecrire une fonction ou procédure qui supprime un auteur de la bibliothèque.

Remarque :

N'oubliez pas de spécifier le rôle du ou des arguments d'une procédure et de leur mode de passage.

Bon courage !