

Exercice 2 (TD arbre)

Déclaration :

Fichier index :

Type Tblocl=structure

Val : tableau[1..N-1] de tcouple

Fils :tableau[1..N] d'entier

Degré :entier

Fin

Type tcouple=structure

Attr :entier

Numbloc :entier

Depl :entier

Fin

FI fichier de Tblocl buffer bufl entete(entier)// entier : numéro du bloc racine

Fichier de données :

Type TblocD=structure

Tab :tableau [1..b] de tenrg

Nb : entier

Fin

Type tenrg=structure

Attr : entier

Champ1 : qlq

Champe2 :qlq

.... Fin

FD fichier de TblocD buffer bufD entete(entier)// entier : nombre de blocs du fichier

Début

lire(v)

Continue → vrai

racine → entete(FI,1)

recherche-inordre(racine,v,continue)

fin

recherche-inordre(racine :entier,v :entier, var continue)

Tanque (continue) faire

Si (racine<> -1)

lireDir(FI, racine, bufl)

pour i allant de 1 à bufl.degre -1 faire

Recherche-inordre(bufl.fils[i],v,continue)

Si bufl.val[i].attr>=v

Continue → faux

Sinon // afficher les éléments a partir du fichier de données)

LireDir(FD, bufD, bufl.val[i].numbloc)

Afficher (bufD.tab[bufD.val[i].depl)

Finpour

Recherche-inordre(bufl.fils[buf.degre],v,continue)

finTQ