**program Ejercicio7;**

cat = ('A'..'D');

type

info = record

id:integer;

autor:string[20];

categ:cat;

end;

arbol = ^nodoAr;

nodoAr = record

dato:info;

hi:arbol;

hd:arbol;

end;

dato2 = record

id:integer;

autor:string[20];

end;

lista2 = ^nodoL2;

nodoL2 = record

dato:dato2;

sig:lista2;

end;

info1 = record

categ:cat;

informes:lista2;

end;

lista = ^nodoL;

nodoL = record

dato:info1;

cant:integer;

sig:lista;

end;

// A

procedure inicializarLista(var l:lista);

var

aux:l;

begin

new(l);

aux:=l;

for i:= 'A' to 'D' do

begin

aux^.dato.categ:= i;

aux^.dato.informes:=nil;

aux^.cant:=0;

aux^.sig:=nil;

aux:=aux^.sig;

new(aux);

end;

end;

procedure agregarEnLista(dato:info; l:lista); //agrego adelante

var

aux:lista2;

begin

while (l^.dato.categ <> dato.categ) do

l:=l^.sig;

l^.cant:=l^.cant + 1;

new(aux);

aux^.sig:= l^.dato.informes;

aux^.dato.id:= dato.id;

aux^.dato.autor:= dato.autor;

l^.dato.informes:= aux;

end;

procedure generarLista(a:arbol; var l:lista);

begin

if (a <> nil) then

begin

agregarEnLista(a^.dato,l); //agrego adelante

generarLista(a^.hi,l);

generarLista(a^.hd,l);

end;

end;

// B

procedure cantidad ( l:lista ; categ: cat);

begin

while (l^.dato.categ <> categ) do

l:= l^.sig;

writeln( 'Cantidad de informes en categoria ',categ,': ',l^.cant);

end;

// C

procedure informar ( a: arbol);

begin

if (a <> nil) then

if ( a^.dato.id > 10) then

if ( a^.dato.id < 40) then

begin

writeln( a^.dato.id );

informar(a^.hi);

informar(a^.hd);

end

else

informar (a^.hi)

else

informar (a^.hd);

end;

// D

procedure pre-order ( a: arbol);

begin

if (a <> nil) then

begin

writeln(a^.dato.id);

pre-order(a^.hi);

pre-order(a^.hd);

end;

end;

procedure inorder ( a: arbol);

begin

if (a <> nil) then

begin

inorder(a^.hi);

writeln(a^.dato.id);

inorder(a^.hd);

end;

end;

procedure post-order ( a: arbol);

begin

if (a <> nil) then

begin

post-order(a^.hi);

post-order(a^.hd);

writeln(a^.dato.id);

end;

end;

// Programa Principal

var

l:lista;

a:arbol;

categ:cat;

begin

inicializarLista(l);

generarLista(a,l);

cantidad (l,categ);

informar (a);

pre-order(a);

inorder(a);

post-order(a);

end.