procedure agregar\_adelante(var l:lista; num:integer);

var

nue:lista;

begin

new(nue);

nue^.dato:=num;

nue^.sig:=l;

l:=nue;

end;

procedure agregar\_atras (var l, ult: lista; num: integer);

var

nue: lista;

begin

new (nue);

nue^.dato:= num;

nue^.sig:= nil;

if (l<> nil) then

ult^.sig:= nue

else

l:= nue;

ult:= nue;

end;

procedure insertar\_ordenado (var l: listaA; r: alumno);

var

ant, nue, act: listaA;

begin

new (nue);

nue^.dato:= r;

act:= l;

ant:= l;

while (act <> nil) and (act^.dato.dni < r.dni) do begin

ant:= act;

act:= act^.sig;

end;

if (ant = act) then

l:= nue

else

ant^.sig:= nue;

nue^.sig:= act;

end;

procedure buscar\_minimo (var v:vector; var min: integer);

var

i,aux: integer;

begin

aux:= 0;

min:= nummin;

for i:= 1 to DimF do

if ( v[i] <> nil) then

if (v[i]^.dato.isbn < min) then begin

min:= v[i]^.dato.isbn;

aux:= i;

end;

if (min <> nummin) then

v[aux]:= v[aux]^.sig;

end;

procedure determinarMinimo(var v: vector; var apeMin: string; var e: empleado);

var

i, lmin: integer;

begin

apeMin:= 'zzz';

for i:= 1 to 4 do

if (v[i] <> NIL) then

if (v[i]^.dato.apellido < apeMin) then begin

apeMin:= v[i]^.dato.apellido;

e:= v[i]^.dato;

lmin:= i;

end;

if (apeMin <> 'zzz') then

v[lmin]:= v[lmin]^.sig;

end;

procedure busquedaDicotomica (v: vector; ini, fin: integer; num: integer; pos: integer);

var

med: integer; //ini:=1; fin,pos:=dimL;

begin

if (pos <> 0) then begin

med:= (ini + fin) div 2;

if (num = v[med]) then

writeln('Lo encontre')

else

if (num < v[med]) then

busquedadicotomica(v, ini, med-1, num, pos-1)

else

busquedadicotomica(v, med+1, fin, num, pos+1);

end

else

writeln('No lo encontre');

end;

program cargar\_vector(var v: vector; var dimL:integer);

var

dato:integer;

begin

dimL:= 0;

read(dato);

while (dato<>99) and (dimL<dimF) do begin

dimL:= dimL + 1;

v[dimL]:= dato;

read(dato);

end;

end;

procedure crearVector4Listas (var v: vector);

var

i, j, num: integer;

begin

for j:= 1 to 4 do

v[j]:= NIL;

for i:= 1 to 4 do begin

num:= random (10);

while (num <> 0)do begin

insertar\_ordenado (v[i], num);

num := random(10);

end;

end;

end;

procedure imprimirDigitos1 (num: integer);

var

aux: integer;

begin

aux:= num;

if ( num <> 0) then begin

aux:= num MOD 10;

num:= num DIV 10;

imprimirDigitos1 (num);

writeln (aux);

end;

end;

procedure imprimirDigitos2 (num: integer);

var

aux: integer;

begin

aux:= num;

if ( num <> 0) then begin

aux:= num MOD 10;

num:= num DIV 10;

writeln (aux);

imprimirDigitos2 (num);

end;

end;

procedure imprimir\_Lista(l:listaA);

begin

while (l<>nil) do begin

writeln('Numero de alumno: ',l^.dato.numA);

writeln();

writeln('Apellido y Nombre: ',l^.dato.ape\_nom);

writeln();

writeln('Dni: ',l^.dato.dni);

writeln();

writeln('Nota final: ',l^.dato.notaF);

l:=l^.sig;

end;

end;

function ContarElementos (l: listaNivel): integer;

var c: integer;

begin

c:= 0;

While (l <> nil) do begin

c:= c+1;

l:= l^.sig;

End;

contarElementos := c;

end;

AGREGARATRAS - Agrega un elemento atras en l}

Procedure AgregarAtras (var l, ult: listaNivel; a:arbol);

var nue:listaNivel;

begin

new (nue);

nue^.info := a;

nue^.sig := nil;

if l= nil then l:= nue

else ult^.sig:= nue;

ult:= nue;

end;

IMPRIMIRPORNIVEL - Muestra los datos del arbol a por niveles

Procedure ImprimirVector ( var vec: vector; dimL: dim );

var

i: dim;

begin

for i:= 1 to dimL do

write ('-----');

writeln;

write (' ');

for i:= 1 to dimL do begin

if(vec[i] < 9)then

write ('0');

write(vec[i], ' | ');

end;

writeln;

for i:= 1 to dimL do

write ('-----');

writeln;

writeln;

End;

procedure imprimirEnOrden (l: lista);

begin

if (l <> nil) then begin

imprimirEnOrden (l^.sig);

writeln (l^.dato);

end;

end;

procedure imprimirEnOrdenInverso (l: lista);

begin

if (l <> nil) then begin

writeln (l^.dato);

imprimirEnOrdenInverso (l^.sig);

end;

end;

procedure Leer\_alumno(var r:alumno);

begin

writeln('Ingrese cantidad de asistencias: ');

readln(r.cantA);

if (r.cantA <> -1) then begin

writeln('Ingrese numero de alumno: ');

readln(r.numA);

writeln('Ingrese nombre y apellido: ');

readln(r.ape\_nom);

writeln('Ingrese dni: ');

readln(r.dni);

writeln('Ingrese nota final: ');

readln(r.notaF);

end;

end;

procedure Ordenacion\_Insercion (var v: vector; DimL: integer);

var

j, i, aux: integer;

begin

for i:= 2 to DimL do begin

aux:= v[i];

j:= i-1;

while (j > 0) and (v[j] > aux) do begin

v[j+1] := v[j];

j:= j -1;

end;

v[j+1]:= aux;

end;

end;