{TERMINADO }

| { ACTIVIDAD 1. Implementar: A. El módulo cargarListaGastos que lea los gastos aleatoriamente (de manera similar a como se generaban los inmuebles) y los almacene ordenados por tipo de gasto. La lectura finaliza cuando se lee monto 0. B. Los módulos cargarGastos(), mergeAcumulador() y minimo (). (sólo deberán transcribirlos de esta presentación). C. Un módulo que reciba la "lista única" y retorne el tipo de gasto con mayor gasto. D. Un programa que invoque a los módulos implementados y compruebe el correcto funcionamiento del mismo.}   program actividad1; uses  GenericLinkedlist;  type gasto=record tipo:string; monto:real; end;  lista=specialize linkedlist <gasto>; vector=array[1..5] of lista;  procedure cargargasto(var a:gasto); var vtipo:array[1..10] of string=('Luz','Telfono','Gas','Seguro','Agua','Alimentos','Ocio','Viajes','Financieros','Alquiler'); begin a.tipo:=vtipo[random(10-1+1)+1]; a.monto:=random(random(1000))\*0.5; end;   // PUNTO A //-----------------------------------------------------------------------------------------------------  procedure insertarordenado(var l:lista;a:gasto); var seguir:boolean; begin seguir:=True; l.reset(); while not l.eol and seguir do begin if a.tipo<l.current().tipo then seguir:=False else l.next(); end; l.insertcurrent(a); end; //-----------------------------------------------------------------------------------------------------  procedure cargarlistagastos(var l:lista); var a:gasto; begin l:=lista.create(); cargargasto(a); while a.monto<>0 do  begin insertarordenado(l,a); cargargasto(a); end; end; //----------------------------------------------------------------------------------------------------- procedure imprimir (var l:lista);begin l.reset(); while not l.eol do  begin write('Tipo de gasto :'); writeln(l.current().tipo);  write('Monto: '); writeln(l.current().monto:2:2); writeln('-------'); l.next(); end; end;  //PUNTO B //cargo vector (persona) con gastos procedure cargarvector (var v:vector); var  i:integer; begin for i:=1 to 5 do  begin cargarlistagastos(v[i]); end; end; procedure minimo (var v:vector;var min:gasto); var posmin:integer; i:integer; begin min.tipo:='zzz';  for i:=1 to 5 do begin if (not v[i].eol()) then  begin  if (v[i].current().tipo<=min.tipo) then  begin  min:=v[i].current();  posmin:=i;  end;  end;  end;  if min.tipo<>'zzz' then begin  v[posmin].next();  end; end;   procedure mergeacumulador(var listaunica:lista; v:vector); var i:integer; min:gasto; gastoactual:gasto; begin listaunica:=lista.create();  for i:=1 to 5 do  begin  v[i].reset();  end; minimo(v,min);  while min.tipo <>'zzz' do begin gastoactual.tipo:=min.tipo; gastoactual.monto:=0; while (min.tipo=gastoactual.tipo) do  begin  gastoactual.monto:=gastoactual.monto+min.monto;  minimo(v,min);  end; listaunica.add(gastoactual); end; end; //-------------------------------// PROGRAMA//--------------------------------------------------------- var v:vector; i:integer; l:lista; begin // PUNTO C randomize; cargarvector(v); for i:=1 to 5 do begin writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'); writeln('Persona ',i); writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'); imprimir(v[i]); end; writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');  writeln('Acumulado'); writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');  mergeacumulador(l,v); imprimir(l);  end. |
| --- |