{TERMINADO }

| {ACTIVIDAD 2 Un centro de distribución recibe paquetes de 6 lugares distintos. De cada paquete se conoce: código de paquete, código de lugar (entre 1 y 6), domicilio de envío y peso del paquete (Entre 1 Kg y 10 Kg en pasos de 500 gramos). Implementar: A. Un módulo llamado LeerPaquete() que lea los datos de paquetes aleatoriamente (de manera similar a como se generaban los inmuebles) y los almacene ordenados por peso y agrupados por código de lugar, en una estructura de datos adecuada. La lectura finaliza cuando se lee el código de paquete -1. Ya está resuelto!!! B. Un módulo que reciba la estructura generada en el punto a y retorne una estructura de datos donde se almacenen las duplas (peso, cantidad) de ese peso, ordenadas por peso. C. Un programa que invoque a los módulos implementados y compruebe el correcto funcionamiento del mismo.}  program distribuicion; uses genericlinkedlist; // --------------------------- DECLARACION DE TIPOS ------------------------------------ type  paquete=record  cod:integer; codlugar:integer; domenvio:string; peso:real; end; acumulador=record peso:real; cantidad:real; end; lista=specialize linkedlist <paquete>; listanueva=specialize linkedlist <acumulador>; vector=array[1..6] of lista;  // --------------------------- MODULOS ------------------------------------   // \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ PUNTO A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  procedure leerpaquete (var a:paquete); var num:string; begin  a.cod:=random(100-1-1)-1; if a.cod <> -1 then begin a.codlugar:=random(6)+1; Str(random(10)+1,num); // CONVIERTE UN NUMERO EN UN STRING - HAY QUE DECALRAR UNA VARIABLE PARA GUARDARLA a.peso:=(random(20-2+1)+2)\*0.5; end; end;   procedure crearvc(var v:vector); var i:integer; begin for i:=1 to 6 do begin v[i]:=lista.create(); end; end;    procedure insertarordenado(var l:lista;valor:paquete); var seguir:boolean; begin l.reset();  seguir:=True; while (not l.eol()) and seguir do begin if valor.peso<=l.current().peso then begin seguir:=False; end else begin l.next(); end; end; l.insertcurrent(valor); end;   procedure cargarlista (var v:vector); var a:paquete; begin crearvc(v); randomize; leerpaquete(a); while a.cod<>-1 do begin insertarordenado(v[a.codlugar],a); leerpaquete(a); end; end;    procedure imprimirlista (l:lista); var a:paquete; begin l.reset(); while (not l.eol()) do begin a:=l.current(); writeln('Codigo Paquete: ',a.cod); writeln('Codigo lugar: ',a.codlugar); writeln('Domicilio Envio: ',a.domenvio); writeln('Peso: ',a.peso:2:2); writeln('------------------------'); l.next(); end; writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'); end;  // \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ PUNTO B \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   { B. Un módulo que reciba la estructura generada en el punto a y retorne una estructura de datos donde se almacenen las duplas (peso, cantidad) de ese peso, ordenadas por peso.}  procedure minimo (var v:vector;var min:paquete); var posmin:integer; i:integer; begin min.cod:=-1; min.peso:=999;  for i:=1 to 6 do begin if (not v[i].eol()) then  begin  if (v[i].current().peso<=min.peso) then  begin  min:=v[i].current();  posmin:=i;  end;  end;  end;  if min.cod<>-1then begin  v[posmin].next();  end; end;   procedure contador (var l2:listanueva;v:vector); var  min:paquete; a:acumulador; i:integer; begin l2:=listanueva.create(); for i:=1 to 6 do   begin  v[i].reset();  end; minimo(v,min);  while min.cod<>-1 do begin a.peso:=min.peso; a.cantidad:=0;  while a.peso=min.peso do  begin  a.cantidad:=a.cantidad+1;  minimo(v,min);  end; l2.add(a) end; end;  procedure imprimirlistanueva (l:listanueva); var a:acumulador; begin l.reset(); while (not l.eol()) do begin a:=l.current(); write('Peso: ',a.peso:10:2); write(' '); writeln('Cantidad total : ',a.cantidad:10:0); l.next(); end; writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'); end;   // \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ PROGRAMA \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  VAR  v:vector; i:integer; l2:listanueva; begin // \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ PUNTO A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ cargarlista(v); writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_IMPRIMO VECTOR POST CARGA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'); for i:=1 to 6 do  begin  writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ VECTOR LUGAR ',i,' \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'); imprimirlista(v[i]); end; contador(l2,v); imprimirlistanueva(l2);  end. |
| --- |