| { 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. B. Un módulo que reciba la estructura generada en el punto a y retorne una estructura de datos donde estén todos los paquetes almacenados y ordenados 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;  lista=specialize linkedlist <paquete>; 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 write('Ingrese codigo de lugar; '); a.codlugar:=random(6)+1; writeln(a.codlugar);  write('Ingrese domicilio de envio: '); Str(random(10)+1,num); // CONVIERTE UN NUMERO EN UN STRING - HAY QUE DECALRAR UNA VARIABLE PARA GUARDARLA a.domenvio:=('Calle ' + num ); // CONCATENA writeln(a.domenvio);  write('Ingrese peso: ');  a.peso:=(random(20-2+1)+2)\*0.5; writeln(a.peso:2:2); writeln('---------------'); 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); writeln('\_\_\_\_\_\_\_\_\_\_DATO RANDOM - LUEGO INGRESADOS POR VECTOR Y ORDENADO \_\_\_\_\_\_\_\_\_\_'); randomize; leerpaquete(a); while a.cod<>-1 do begin insertarordenado(v[a.codlugar],a); leerpaquete(a); end; writeln('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_'); 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 MERGE \_\_\_\_\_\_\_\_\_\_\_\_\_\_     procedure minimo (var v:vector;var min:paquete;var posmin:integer;i:integer);    begin  if i<=6 then 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;  minimo (v,min,posmin,i+1);  end  else if min.peso<>999 then   begin  v[posmin].next();  end; end;   procedure merge(v:vector;var listanueva:lista); var i,posmin:integer; min:paquete; begin listanueva:=lista.create();  for i:=1 to 6 do  begin v[i].reset(); end;   min.peso:=999; posmin:=1; minimo(v,min,posmin,1); while (min.peso<>999) do  begin listanueva.add(min); min.peso:=999; posmin:=1;  minimo(v,min,posmin,1); end; end;     // \_\_\_\_\_\_\_\_\_\_\_ PROGRAMA \_\_\_\_\_\_\_\_\_\_\_\_\_\_  VAR listanueva:lista; v:vector; i:integer; 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; // \_\_\_\_\_\_\_\_\_\_\_ PUNTO B MERGE \_\_\_\_\_\_\_\_\_\_\_\_\_\_  merge(v,listanueva); writeln('\_\_\_\_\_\_\_\_\_\_IMPRIMO LISTA MERGE\_\_\_\_\_\_\_\_\_\_'); imprimirlista(listanueva); end. |
| --- |