**Trabajo Práctico N° 3:**

**Registros. Ejercicios con Corte de Control.**

**Ejercicio 1.**

*Dado el siguiente programa:*

program TP3\_E1;

*{$codepage UTF8}*

uses crt;

type

  str20=string[20];

  alumno=record

    codigo: integer;

    nombre: str20;

    promedio: real;

end;

procedure leer(var alu: alumno);

begin

  write('Ingrese el código del alumno'); read(alu.codigo);

  if (alu.codigo<>0) then

  begin

write('Ingrese el nombre del alumno'); read(alu.nombre);

write('Ingrese el promedio del alumno'); read(alu.promedio);

  end;

end;

var

  a: alumno;

begin

end.

**(a)** *Completar el programa principal para que lea información de alumnos (código, nombre, promedio) e informe la cantidad de alumnos leídos. La lectura finaliza cuando ingresa un alumno con código 0, que no debe procesarse. Nota: Utilizar el módulo leer.*

program TP3\_E1a;

*{$codepage UTF8}*

uses crt;

const

  codigo\_salida=0;

type

  str20=string[20];

  alumno=record

    codigo: integer;

    nombre: str20;

    promedio: real;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer(var alu: alumno);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    alu.codigo:=codigo\_salida

  else

    alu.codigo:=random(high(int16));

  if (alu.codigo<>codigo\_salida) then

  begin

    alu.nombre:=random\_string(5+random(6));

    alu.promedio:=1+random(91)/10;

  end;

end;

var

  a: alumno;

  alumnos\_leidos: integer;

begin

  randomize;

  alumnos\_leidos:=0;

  leer(a);

  while (a.codigo<>codigo\_salida) do

  begin

    alumnos\_leidos:=alumnos\_leidos+1;

    leer(a);

  end;

  textcolor(green); write('La cantidad de alumnos leídos es '); textcolor(red); write(alumnos\_leidos);

end.

**(b)** *Modificar al programa anterior para que, al finalizar la lectura de todos los alumnos, se informe también el nombre del alumno con mejor promedio.*

program TP3\_E1b;

*{$codepage UTF8}*

uses crt;

const

  codigo\_salida=0;

type

  str20=string[20];

  alumno=record

    codigo: integer;

    nombre: str20;

    promedio: real;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer(var alu: alumno);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    alu.codigo:=codigo\_salida

  else

    alu.codigo:=random(high(int16));

  if (alu.codigo<>codigo\_salida) then

  begin

    alu.nombre:=random\_string(5+random(6));

    alu.promedio:=1+random(91)/10;

  end;

end;

procedure actualizar\_maximo(promedio: real; nombre: str20; var promedio\_max: real; var nombre\_max: str20);

begin

  if (promedio>promedio\_max) then

  begin

    promedio\_max:=promedio;

    nombre\_max:=nombre;

  end;

end;

var

  a: alumno;

  alumnos\_leidos: integer;

  promedio\_max: real;

  nombre\_max: str20;

begin

  randomize;

  alumnos\_leidos:=0;

  leer(a);

  while (a.codigo<>codigo\_salida) do

  begin

    alumnos\_leidos:=alumnos\_leidos+1;

    actualizar\_maximo(a.promedio,a.nombre,promedio\_max,nombre\_max);

    leer(a);

  end;

  textcolor(green); write('La cantidad de alumnos leidos es '); textcolor(red); writeln(alumnos\_leidos);

  textcolor(green); write('El nombre del alumno con mejor promedio es '); textcolor(red); write(nombre\_max);

end.

**Ejercicio 2.**

*El registro civil de La Plata ha solicitado un programa para analizar la distribución de casamientos durante el año 2019. Para ello, cuenta con información de las fechas de todos los casamientos realizados durante ese año.*

**(a)** *Analizar y definir un tipo de dato adecuado para almacenar la información de la fecha de cada casamiento.*

**(b)** *Implementar un módulo que lea una fecha desde teclado y la retorne en un parámetro cuyo tipo es el definido en el inciso (a).*

**(c)** *Implementar un programa que:*

* *Lea la fecha de todos los casamientos realizados en 2019. La lectura finaliza al ingresar el año 2020, que no debe procesarse.*
* *Informe la cantidad de casamientos realizados durante los meses de verano (enero, febrero y marzo) y la cantidad de casamientos realizados en los primeros 10 días de cada mes. Nota: Utilizar el módulo realizado en (b) para la lectura de fecha.*

program TP3\_E2;

*{$codepage UTF8}*

uses crt;

const

  anio\_salida=2020;

  dia\_ini=1; dia\_fin=31;

  mes\_ini=1; mes\_fin=12;

  mes\_corte1=1; mes\_corte2=2; mes\_corte3=3;

  dia\_corte=10;

type

  t\_dia=dia\_ini..dia\_fin;

  t\_mes=mes\_ini..mes\_fin;

  t\_registro\_casamiento=record

    dia: t\_dia;

    mes: t\_mes;

    anio: int16;

  end;

procedure leer\_casamiento(var registro\_casamiento: t\_registro\_casamiento);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_casamiento.anio:=anio\_salida

  else

    registro\_casamiento.anio:=2019;

  if (registro\_casamiento.anio<>anio\_salida) then

  begin

    registro\_casamiento.dia:=dia\_ini+random(dia\_fin);

    registro\_casamiento.mes:=mes\_ini+random(mes\_fin);

  end;

end;

procedure leer\_casamientos(var casamientos\_corte\_mes, casamientos\_corte\_dia: int16);

var

  registro\_casamiento: t\_registro\_casamiento;

begin

  leer\_casamiento(registro\_casamiento);

  while (registro\_casamiento.anio<>anio\_salida) do

  begin

    if ((registro\_casamiento.mes=mes\_corte1) or (registro\_casamiento.mes=mes\_corte2) or (registro\_casamiento.mes=mes\_corte3)) then

      casamientos\_corte\_mes:=casamientos\_corte\_mes+1;

    if (registro\_casamiento.dia<=dia\_corte) then

      casamientos\_corte\_dia:=casamientos\_corte\_dia+1;

    leer\_casamiento(registro\_casamiento);

  end;

end;

var

  casamientos\_corte\_mes, casamientos\_corte\_dia: int16;

begin

  randomize;

  casamientos\_corte\_mes:=0;

  casamientos\_corte\_dia:=0;

  leer\_casamientos(casamientos\_corte\_mes,casamientos\_corte\_dia);

  textcolor(green); write('La cantidad de casamientos realizados durante los meses de verano (enero, febrero y marzo) es '); textcolor(red); writeln(casamientos\_corte\_mes);

  textcolor(green); write('La cantidad de casamientos realizados en los primeros '); textcolor(yellow); write(dia\_corte); textcolor(green); write(' días de cada mes es '); textcolor(red); write(casamientos\_corte\_dia);

end.

**Ejercicio 3.**

*El Ministerio de Educación desea realizar un relevamiento de las 2400 escuelas primarias de la provincia de Bs. As., con el objetivo de evaluar si se cumple la proporción de alumnos por docente calculada por la UNESCO para el año 2015 (1 docente cada 23,435 alumnos). Para ello, se cuenta con información de: CUE (código único de establecimiento), nombre del establecimiento, cantidad de docentes, cantidad de alumnos, localidad. Se pide implementar un programa que procese la información y determine:*

* *Cantidad de escuelas de La Plata con una relación de alumnos por docente superior a la sugerida por UNESCO.*
* *CUE y nombre de las dos escuelas con mejor relación entre docentes y alumnos.*

*El programa debe utilizar:*

* *Un módulo para la lectura de la información de la escuela.*
* *Un módulo para determinar la relación docente-alumno (esa relación se obtiene del cociente entre la cantidad de alumnos y la cantidad de docentes).*

program TP3\_E3;

*{$codepage UTF8}*

uses crt;

const

  escuelas\_total=2400;

  localidad\_corte='La Plata'; ratio\_corte=23.435;

type

  t\_registro\_escuela=record

    escuela: int16;

    nombre: string;

    docentes: int16;

    alumnos: int16;

    localidad: string;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_escuela(var registro\_escuela: t\_registro\_escuela);

var

  i: int8;

begin

  registro\_escuela.escuela:=1+random(high(int16));

  registro\_escuela.nombre:='Escuela '+random\_string(5+random(6));

  registro\_escuela.docentes:=1+random(100);

  registro\_escuela.alumnos:=1+random(1000);

  i:=random(100);

  if (i=0) then

    registro\_escuela.localidad:=localidad\_corte

  else

    registro\_escuela.localidad:='Localidad '+random\_string(5+random(6));

end;

function ratio\_alumnos\_docente(registro\_escuela: t\_registro\_escuela): real;

begin

    ratio\_alumnos\_docente:=registro\_escuela.alumnos/registro\_escuela.docentes;

end;

procedure actualizar\_minimos(ratio: real; registro\_escuela: t\_registro\_escuela; var ratio\_min1, ratio\_min2: real; var escuela\_min1, escuela\_min2: int16; var nombre\_min1, nombre\_min2: string);

begin

  if (ratio<ratio\_min1) then

  begin

    ratio\_min2:=ratio\_min1;

    escuela\_min2:=escuela\_min1;

    nombre\_min2:=nombre\_min1;

    nombre\_min1:=registro\_escuela.nombre;

    escuela\_min1:=registro\_escuela.escuela;

  end

  else

    if (ratio<ratio\_min2) then

    begin

      ratio\_min2:=ratio;

      escuela\_min2:=registro\_escuela.escuela;

      nombre\_min2:=registro\_escuela.nombre;

  end;

end;

procedure leer\_escuelas(var escuelas\_corte, escuela\_min1, escuela\_min2: int16; var nombre\_min1, nombre\_min2: string);

var

  registro\_escuela: t\_registro\_escuela;

  i: int16;

  ratio, ratio\_min1, ratio\_min2: real;

begin

  ratio:=0;

  ratio\_min1:=9999999; ratio\_min2:=9999999;

  for i:= 1 to escuelas\_total do

  begin

    leer\_escuela(registro\_escuela);

    ratio:=ratio\_alumnos\_docente(registro\_escuela);

    actualizar\_minimos(ratio,registro\_escuela,ratio\_min1,ratio\_min2,escuela\_min1,escuela\_min2,nombre\_min1,nombre\_min2);

    if ((registro\_escuela.localidad=localidad\_corte) and (ratio>ratio\_corte)) then

      escuelas\_corte:=escuelas\_corte+1;

  end;

end;

var

  escuelas\_corte, escuela\_min1, escuela\_min2: int16;

  nombre\_min1, nombre\_min2: string;

begin

  randomize;

  escuelas\_corte:=0;

  escuela\_min1:=0; escuela\_min2:=0; nombre\_min1:=''; nombre\_min2:='';

  leer\_escuelas(escuelas\_corte,escuela\_min1,escuela\_min2,nombre\_min1,nombre\_min2);

  textcolor(green); write('La cantidad de escuelas de '); textcolor(yellow); write(localidad\_corte); textcolor(green); write(' con una relación de alumnos por docente superior a la sugerida por UNESCO ('); textcolor(yellow); write(ratio\_corte:0:2); textcolor(green); write(') es '); textcolor(red); writeln(escuelas\_corte);

  textcolor(green); write('Los CUEs de las dos escuelas con mejor relación entre docentes y alumnos son '); textcolor(red); write(escuela\_min1); textcolor(green); write(' y '); textcolor(red); writeln(escuela\_min2);

  textcolor(green); write('Los nombres de las dos escuelas con mejor relación entre docentes y alumnos son '); textcolor(red); write(nombre\_min1); textcolor(green); write(' y '); textcolor(red); write(nombre\_min2);

end.

**Ejercicio 4.**

*Una compañía de telefonía celular debe realizar la facturación mensual de sus 9300 clientes con planes de consumo ilimitados (clientes que pagan por lo que consumen). Para cada cliente, se conoce su código de cliente y cantidad de líneas a su nombre. De cada línea, se tiene el número de teléfono, la cantidad de minutos consumidos y la cantidad de MB consumidos en el mes. Se pide implementar un programa que lea los datos de los clientes de la compañía e informe el monto total a facturar para cada uno. Para ello, se requiere:*

* *Realizar un módulo que lea la información de una línea de teléfono.*
* *Realizar un módulo que reciba los datos de un cliente, lea la información de todas sus líneas (utilizando el módulo desarrollado en el inciso (a)) y retorne la cantidad total de minutos y la cantidad total de MB a facturar del cliente.*

*Nota: Para realizar los cálculos tener en cuenta que cada minuto cuesta $3,40 y cada MB consumido cuesta $1,35.*

program TP3\_E4;

*{$codepage UTF8}*

uses crt;

const

  clientes\_total=9300;

  costo\_minuto=3.40; costo\_MB=1.35;

type

  t\_cliente=1..clientes\_total;

  t\_registro\_cliente=record

    cliente: int16;

    lineas: int8;

  end;

  t\_registro\_linea=record

    numero: int16;

    minutos: int16;

    MBs: int16;

  end;

procedure leer\_cliente(var registro\_cliente: t\_registro\_cliente);

begin

  registro\_cliente.cliente:=1+random(high(int16));

  registro\_cliente.lineas:=1+random(5);

end;

procedure leer\_linea(var registro\_linea: t\_registro\_linea);

begin

  registro\_linea.numero:=1+random(high(int16));

  registro\_linea.minutos:=1+random(10000);

  registro\_linea.MBs:=1+random(10000);

end;

procedure leer\_lineas(lineas: int8; var minutos\_cliente, MBs\_cliente: int32);

var

  registro\_linea: t\_registro\_linea;

  i: int8;

begin

  for i:= 1 to lineas do

    begin

      leer\_linea(registro\_linea);

      minutos\_cliente:=minutos\_cliente+registro\_linea.minutos;

      MBs\_cliente:=MBs\_cliente+registro\_linea.MBs;

    end;

end;

var

  registro\_cliente: t\_registro\_cliente;

  i: t\_cliente;

  minutos\_cliente, MBs\_cliente: int32;

  monto\_cliente: real;

begin

  randomize;

  for i:= 1 to clientes\_total do

  begin

    minutos\_cliente:=0; MBs\_cliente:=0;

    leer\_cliente(registro\_cliente);

    leer\_lineas(registro\_cliente.lineas,minutos\_cliente,MBs\_cliente);

    monto\_cliente:=minutos\_cliente\*costo\_minuto+MBs\_cliente\*costo\_MB;

    textcolor(green); write('El monto total a facturar del cliente '); textcolor(yellow); write(i); textcolor(green); write(' es $'); textcolor(red); writeln(monto\_cliente:0:2);

  end;

end.

**Ejercicio 5.**

*Realizar un programa que lea información de autos que están a la venta en una concesionaria. De cada auto, se lee: marca, modelo y precio. La lectura finaliza cuando se ingresa la marca “ZZZ”, que no debe procesarse. La información se ingresa ordenada por marca. Se pide calcular e informar:*

* *El precio promedio por marca.*
* *Marca y modelo del auto más caro.*

program TP3\_E5;

*{$codepage UTF8}*

uses crt;

const

  marca\_salida='ZZZ';

type

  t\_registro\_auto=record

    marca: string;

    modelo: string;

    precio: real;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_auto(var registro\_auto: t\_registro\_auto; marca: string);

var

  i: int8;

begin

  i:=random(101);

  if (i=0) then

    registro\_auto.marca:=marca\_salida

  else if (i<=50) then

    registro\_auto.marca:=marca

  else

    registro\_auto.marca:='Marca '+random\_string(5+random(6));

  if (registro\_auto.marca<>marca\_salida) then

  begin

    registro\_auto.modelo:='Modelo '+random\_string(5+random(6));

    registro\_auto.precio:=1000+random(99001);

  end;

end;

procedure actualizar\_maximos(registro\_auto: t\_registro\_auto; var precio\_max: real; var marca\_max, modelo\_max: string);

begin

  if (registro\_auto.precio>precio\_max) then

  begin

    precio\_max:=registro\_auto.precio;

    marca\_max:=registro\_auto.marca;

    modelo\_max:=registro\_auto.modelo;

  end;

end;

procedure leer\_autos(var marca\_max, modelo\_max: string);

var

  registro\_auto: t\_registro\_auto;

  autos\_total: int32;

  precio\_total, precio\_prom, precio\_max: real;

  marca: string;

begin

  precio\_max:=-9999999;

  marca:='Marca XXX';

  leer\_auto(registro\_auto,marca);

  while (registro\_auto.marca<>marca\_salida) do

  begin

    marca:=registro\_auto.marca;

    precio\_total:=0; autos\_total:=0; precio\_prom:=0;

    while ((registro\_auto.marca<>marca\_salida) and (registro\_auto.marca=marca)) do

    begin

      precio\_total:=precio\_total+registro\_auto.precio;

      autos\_total:=autos\_total+1;

      actualizar\_maximos(registro\_auto,precio\_max,marca\_max,modelo\_max);

      leer\_auto(registro\_auto,marca);

    end;

    precio\_prom:=precio\_total/autos\_total;

    textcolor(green); write('El precio promedio de la marca '); textcolor(red); write(marca); textcolor(green); write(' es '); textcolor(red); writeln(precio\_prom:0:2);

  end;

end;

var

  marca\_max, modelo\_max: string;

begin

  randomize;

  marca\_max:=''; modelo\_max:='';

  leer\_autos(marca\_max,modelo\_max);

  textcolor(green); write('La marca y el modelo del auto más caro son '); textcolor(red); write(marca\_max); textcolor(green); write(' y '); textcolor(red); write(modelo\_max); textcolor(green); write(', respectivamente');

end.

**Ejercicio 6.**

*Una empresa importadora de microprocesadores desea implementar un sistema de software para analizar la información de los productos que mantiene, actualmente, en stock. Para ello, se conoce la siguiente información de los microprocesadores: marca (Intel, AMD, NVidia, etc.), línea (Xeon, Core i7, Opteron, Atom, Centrino, etc.), cantidad de cores o núcleos de procesamiento (1, 2, 4, 8), velocidad del reloj (medida en Ghz) y tamaño en nanómetros (nm) de los transistores (14, 22, 32, 45, etc.). La información de los microprocesadores se lee de forma consecutiva por marca de procesador y la lectura finaliza al ingresar un procesador con 0 cores (que no debe procesarse). Se pide implementar un programa que lea información de los microprocesadores de la empresa importadora e informe:*

* *Marca y línea de todos los procesadores de más de 2 cores con transistores de, a lo sumo, 22 nm.*
* *Las dos marcas con mayor cantidad de procesadores con transistores de 14 nm.*
* *Cantidad de procesadores multicore (de más de un core) de Intel o AMD, cuyos relojes alcancen velocidades de, al menos, 2 Ghz.*

program TP3\_E6;

*{$codepage UTF8}*

uses crt;

const

  cores\_salida=0;

  cores\_corte=2;

  transistores\_corte1=22;

  transistores\_corte2=14;

  marca\_corte1='Intel'; marca\_corte2='AMD'; velocidad\_corte=2.0;

type

  t\_registro\_procesador=record

    cores: int16;

    marca: string;

    linea: string;

    velocidad: real;

    transistores: int16;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_procesador(var registro\_procesador: t\_registro\_procesador; marca: string; var marca\_pos: int8);

var

  vector\_cores: array[1..4] of int8=(1, 2, 4, 8);

  vector\_marcas: array[1..3] of string=('Intel', 'AMD', 'Nvidia');

  vector\_lineas: array[1..5] of string=('Xeon', 'Core i7', 'Opteron', 'Atom', 'Centrino');

  vector\_nms: array[1..4] of int8=(14, 22, 32, 45);

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_procesador.cores:=cores\_salida

  else

    registro\_procesador.cores:=vector\_cores[1+random(4)];

  if (registro\_procesador.cores<>cores\_salida) then

  begin

    i:=random(10);

    if (i=0) then

    begin

      marca\_pos:=marca\_pos+1;

      if (marca\_pos<=3) then

        registro\_procesador.marca:=vector\_marcas[marca\_pos]

      else

        registro\_procesador.marca:=random\_string(5+random(6));

    end

    else

      if (marca\_pos<=3) then

        registro\_procesador.marca:=vector\_marcas[marca\_pos]

      else

        registro\_procesador.marca:=marca;

    registro\_procesador.linea:=vector\_lineas[1+random(5)];

    registro\_procesador.velocidad:=1+random(40)/10;

    registro\_procesador.transistores:=vector\_nms[1+random(4)];

  end;

end;

procedure actualizar\_maximos(transistores\_marca: int16; marca: string; var transistores\_max1, transistores\_max2: int16; var marca\_max1, marca\_max2: string);

begin

  if (transistores\_marca>transistores\_max1) then

  begin

    transistores\_max2:=transistores\_max1;

    marca\_max2:=marca\_max1;

    transistores\_max1:=transistores\_marca;

    marca\_max1:=marca;

  end

  else

    if (transistores\_marca>transistores\_max2) then

    begin

      transistores\_max2:=transistores\_marca;

      marca\_max2:=marca;

    end;

end;

procedure leer\_procesadores(var procesadores\_corte: int16; var marca\_max1, marca\_max2: string);

var

  registro\_procesador: t\_registro\_procesador;

  marca\_pos: int8;

  transistores\_marca, transistores\_max1, transistores\_max2: int16;

  marca: string;

begin

  transistores\_max1:=low(int16); transistores\_max2:=low(int16);

  marca:=''; marca\_pos:=1;

  leer\_procesador(registro\_procesador,marca,marca\_pos);

  while (registro\_procesador.cores<>cores\_salida) do

  begin

    marca:=registro\_procesador.marca;

    transistores\_marca:=0;

    while ((registro\_procesador.cores<>cores\_salida) and (registro\_procesador.marca=marca)) do

    begin

      if ((registro\_procesador.cores>cores\_corte) and (registro\_procesador.transistores<=transistores\_corte1)) then

      begin

        textcolor(green); write('La marca y la línea de este procesador con más de '); textcolor(yellow); write(cores\_corte); textcolor(green); write(' cores con transistores de, a lo sumo, '); textcolor(yellow); write(transistores\_corte1); textcolor(green); write(' nm. son '); textcolor(red); write(registro\_procesador.marca); textcolor(green); write(' y '); textcolor(red); write(registro\_procesador.linea); textcolor(green); writeln(', respectivamente');

      end;

      if (registro\_procesador.transistores=transistores\_corte2) then

        transistores\_marca:=transistores\_marca+1;

      if ((registro\_procesador.cores>=cores\_corte) and ((registro\_procesador.marca=marca\_corte1) or (registro\_procesador.marca=marca\_corte2)) and (registro\_procesador.velocidad>=velocidad\_corte)) then

        procesadores\_corte:=procesadores\_corte+1;

      leer\_procesador(registro\_procesador,marca,marca\_pos);

    end;

    actualizar\_maximos(transistores\_marca,marca,transistores\_max1,transistores\_max2,marca\_max1,marca\_max2);

  end;

end;

var

  procesadores\_corte: int16;

  marca\_max1, marca\_max2: string;

begin

  randomize;

  marca\_max1:=''; marca\_max2:='';

  procesadores\_corte:=0;

  leer\_procesadores(procesadores\_corte,marca\_max1,marca\_max2);

  textcolor(green); write('Las dos marcas con mayor cantidad de procesadores con transistores de '); textcolor(yellow); write(transistores\_corte2); textcolor(green); write(' nm. son '); textcolor(red); write(marca\_max1); textcolor(green); write(' y '); textcolor(red); writeln(marca\_max2);

  textcolor(green); write('La cantidad de procesadores multicore (de más de un core) de '); textcolor(yellow); write(marca\_corte1); textcolor(green); write(' o '); textcolor(yellow); write(marca\_corte2); textcolor(green); write(' cuyos relojes alcancen velocidades de, al menos, '); textcolor(yellow); write(velocidad\_corte:0:2); textcolor(green); write(' Ghz es '); textcolor(red); write(procesadores\_corte);

end.

**Ejercicio 7.**

*Realizar un programa que lea información de centros de investigación de Universidades Nacionales. De cada centro, se lee su nombre abreviado (ej., LIDI, LIFIA, LINTI), la universidad a la que pertenece, la cantidad de investigadores y la cantidad de becarios que poseen. La información se lee de forma consecutiva por universidad y la lectura finaliza al leer un centro con 0 investigadores, que no debe procesarse. Informar:*

* *Cantidad total de centros para cada universidad.*
* *Universidad con mayor cantidad de investigadores en sus centros.*
* *Los dos centros con menor cantidad de becarios.*

program TP3\_E7;

*{$codepage UTF8}*

uses crt;

const

  investigadores\_salida=0;

type

  t\_registro\_centro=record

    centro: string;

    universidad: string;

    investigadores: int16;

    becarios: int16;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_centro(var registro\_centro: t\_registro\_centro; universidad: string);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_centro.investigadores:=investigadores\_salida

  else

    registro\_centro.investigadores:=1+random(100);

  if (registro\_centro.investigadores<>investigadores\_salida) then

  begin

    registro\_centro.centro:='Centro '+random\_string(5+random(6));

    i:=random(2);

    if (i=0) then

      registro\_centro.universidad:=universidad

    else

      registro\_centro.universidad:='Universidad '+random\_string(5+random(6));

    registro\_centro.becarios:=1+random(100);

  end;

end;

procedure actualizar\_minimos(becarios: int16; centro: string; var becarios\_min1, becarios\_min2: int16; var centro\_min1, centro\_min2: string);

begin

  if (becarios<becarios\_min1) then

  begin

    becarios\_min2:=becarios\_min1;

    centro\_min2:=centro\_min1;

    becarios\_min1:=becarios;

    centro\_min1:=centro;

  end

  else

    if (becarios<becarios\_min2) then

    begin

      becarios\_min2:=becarios;

      centro\_min2:=centro;

    end;

end;

procedure actualizar\_maximo(investigadores\_universidad: int16; universidad: string; var investigadores\_max: int16; var universidad\_max: string);

begin

  if (investigadores\_universidad>investigadores\_max) then

  begin

    investigadores\_max:=investigadores\_universidad;

    universidad\_max:=universidad;

  end;

end;

procedure leer\_centros(var universidad\_max, centro\_min1, centro\_min2: string);

var

  registro\_centro: t\_registro\_centro;

  centros\_universidad, investigadores\_universidad, investigadores\_max, becarios\_min1, becarios\_min2: int16;

  universidad: string;

begin

  investigadores\_max:=low(int16);

  becarios\_min1:=high(int16); becarios\_min2:=high(int16);

  universidad:='Universidad XXX';

  leer\_centro(registro\_centro,universidad);

  while (registro\_centro.investigadores<>investigadores\_salida) do

  begin

    universidad:=registro\_centro.universidad;

    centros\_universidad:=0;

    investigadores\_universidad:=0;

    while ((registro\_centro.investigadores<>investigadores\_salida) and (registro\_centro.universidad=universidad)) do

    begin

      centros\_universidad:=centros\_universidad+1;

      investigadores\_universidad:=investigadores\_universidad+registro\_centro.investigadores;

      actualizar\_minimos(registro\_centro.becarios,registro\_centro.centro,becarios\_min1,becarios\_min2,centro\_min1,centro\_min2);

      leer\_centro(registro\_centro,universidad);

    end;

    textcolor(green); write('La cantidad total de centros de la universidad '); textcolor(yellow); write(universidad); textcolor(green); write(' es '); textcolor(red); writeln(centros\_universidad);

    actualizar\_maximo(investigadores\_universidad,universidad,investigadores\_max,universidad\_max);

  end;

end;

var

  universidad\_max, centro\_min1, centro\_min2: string;

begin

  randomize;

  universidad\_max:='';

  centro\_min1:=''; centro\_min2:='';

  leer\_centros(universidad\_max,centro\_min1,centro\_min2);

  textcolor(green); write('La universidad con mayor cantidad de investigadores en sus centros es '); textcolor(red); writeln(universidad\_max);

  textcolor(green); write('Los dos centros con menor cantidad de becarios son '); textcolor(red); write(centro\_min1); textcolor(green); write(' y '); textcolor(red); write(centro\_min2);

end.

**Ejercicio 8.**

*La Comisión Provincial por la Memoria desea analizar la información de los proyectos presentados en el programa Jóvenes y Memoria durante la convocatoria 2020. Cada proyecto, posee un código único, un título, el docente coordinador (DNI, nombre y apellido, email), la cantidad de alumnos que participan del proyecto, el nombre de la escuela y la localidad a la que pertenece. Cada escuela puede presentar más de un proyecto. La información se ingresa ordenada consecutivamente por localidad y, para cada localidad, por escuela. Realizar un programa que lea la información de los proyectos hasta que se ingrese el proyecto con código -1 (que no debe procesarse) e informe:*

* *Cantidad total de escuelas que participan en la convocatoria 2020 y cantidad de escuelas por cada localidad.*
* *Nombres de las dos escuelas con mayor cantidad de alumnos participantes.*
* *Título de los proyectos de la localidad de Daireaux cuyo código posee igual cantidad de dígitos pares e impares.*

program TP3\_E8;

*{$codepage UTF8}*

uses crt;

const

  proyecto\_salida=-1;

  localidad\_corte='Daireaux';

type

  t\_registro\_docente=record

    dni: int32;

    nombre: string;

    apellido: string;

    email: string;

  end;

  t\_registro\_proyecto=record

    proyecto: int16;

    titulo: string;

    docente: t\_registro\_docente;

    alumnos: int16;

    escuela: string;

    localidad: string;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_docente(var registro\_docente: t\_registro\_docente);

var

  vector\_emails: array[1..3] of string=('@gmail.com', '@hotmail.com', '@yahoo.com');

begin

  registro\_docente.dni:=10000000+random(30000001);

  registro\_docente.nombre:=random\_string(5+random(6));

  registro\_docente.apellido:=random\_string(5+random(6));

  registro\_docente.email:=random\_string(5+random(6))+vector\_emails[1+random(3)];

end;

procedure leer\_proyecto(var registro\_proyecto: t\_registro\_proyecto; localidad, escuela: string);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_proyecto.proyecto:=proyecto\_salida

  else

    registro\_proyecto.proyecto:=1+random(high(int16));

  if (registro\_proyecto.proyecto<>proyecto\_salida) then

  begin

    registro\_proyecto.titulo:='Proyecto '+random\_string(5+random(6));

    leer\_docente(registro\_proyecto.docente);

    registro\_proyecto.alumnos:=1+random(100);

    i:=random(2);

    if (i=0) then

      registro\_proyecto.localidad:=localidad

    else

      registro\_proyecto.localidad:='Localidad '+random\_string(5+random(6));

    i:=random(2);

    if (i=0) then

      registro\_proyecto.escuela:=escuela

    else

      registro\_proyecto.escuela:='Escuela '+random\_string(5+random(6));

  end;

end;

procedure actualizar\_maximos(alumnos: int16; escuela: string; var alumnos\_max1, alumnos\_max2: int16; var escuela\_max1, escuela\_max2: string);

begin

  if (alumnos>alumnos\_max1) then

  begin

    alumnos\_max2:=alumnos\_max1;

    escuela\_max2:=escuela\_max1;

    alumnos\_max1:=alumnos;

    escuela\_max1:=escuela;

  end

  else

    if (alumnos>alumnos\_max2) then

    begin

      alumnos\_max2:=alumnos;

      escuela\_max2:=escuela;

    end;

end;

function contar\_pares\_impares(proyecto: int16): boolean;

var

  pares, impares: int16;

begin

  pares:=0; impares:=0;

  while (proyecto<>0) do

  begin

    if (proyecto mod 2=0) then

      pares:=pares+1

    else

      impares:=impares+1;

    proyecto:=proyecto div 10;

  end;

  if (pares=impares) then

    contar\_pares\_impares:=true

  else

    contar\_pares\_impares:=false;

end;

procedure leer\_proyectos(var escuelas\_total: int16; var escuela\_max1, escuela\_max2: string);

var

  registro\_proyecto: t\_registro\_proyecto;

  escuelas\_localidad, alumnos\_escuela, alumnos\_max1, alumnos\_max2: int16;

  localidad, escuela: string;

begin

  alumnos\_max1:=low(int16); alumnos\_max2:=low(int16);

  localidad:=localidad\_corte; escuela:='Escuela XXX';

  leer\_proyecto(registro\_proyecto,localidad,escuela);

  while (registro\_proyecto.proyecto<>proyecto\_salida) do

  begin

    localidad:=registro\_proyecto.localidad;

    escuelas\_localidad:=0;

    while ((registro\_proyecto.proyecto<>proyecto\_salida) and (registro\_proyecto.localidad=localidad)) do

    begin

      escuela:=registro\_proyecto.escuela;

      escuelas\_localidad:=escuelas\_localidad+1;

      alumnos\_escuela:=0;

      while ((registro\_proyecto.proyecto<>proyecto\_salida) and (registro\_proyecto.localidad=localidad) and (registro\_proyecto.escuela=escuela)) do

      begin

        alumnos\_escuela:=alumnos\_escuela+registro\_proyecto.alumnos;

        if ((registro\_proyecto.localidad=localidad\_corte) and (contar\_pares\_impares(registro\_proyecto.proyecto)=true)) then

        begin

          textcolor(green); write('El título de este proyecto de la localidad '); textcolor(yellow); write(localidad\_corte); textcolor(green); write(', cuyo código posee igual cantidad de dígitos pares e impares, es '); textcolor(red); writeln(registro\_proyecto.titulo);

        end;

        leer\_proyecto(registro\_proyecto,localidad,escuela);

      end;

      actualizar\_maximos(alumnos\_escuela,escuela,alumnos\_max1,alumnos\_max2,escuela\_max1,escuela\_max2);

    end;

    escuelas\_total:=escuelas\_total+escuelas\_localidad;

    textcolor(green); write('La cantidad de escuelas de la localidad '); textcolor(red); write(localidad); textcolor(green); write(' es '); textcolor(red); writeln(escuelas\_localidad);

  end;

end;

var

  escuelas\_total: int16;

  escuela\_max1, escuela\_max2: string;

begin

  randomize;

  escuelas\_total:=0;

  escuela\_max1:=''; escuela\_max2:='';

  leer\_proyectos(escuelas\_total,escuela\_max1,escuela\_max2);

  textcolor(green); write('La cantidad total de escuelas que participan en la convocatoria 2020 es '); textcolor(red); writeln(escuelas\_total);

  textcolor(green); write('Los nombres de las dos escuelas con mayor cantidad de alumnos participantes son '); textcolor(red); write(escuela\_max1); textcolor(green); write(' y '); textcolor(red); write(escuela\_max2);

end.

**Ejercicio 9.**

*Realizar un programa que lea información de los candidatos ganadores de las últimas elecciones a intendente de la provincia de Buenos Aires. Para cada candidato, se lee: localidad, apellido del candidato, cantidad de votos obtenidos y cantidad de votantes de la localidad. La lectura finaliza al leer la localidad “Zárate”, que debe procesarse. Informar:*

* *El intendente que obtuvo la mayor cantidad de votos en la elección.*
* *El intendente que obtuvo el mayor porcentaje de votos de la elección.*

program TP3\_E9;

*{$codepage UTF8}*

uses crt;

const

  localidad\_salida='Zarate';

type

  t\_registro\_candidato=record

    localidad: string;

    apellido: string;

    votos: int16;

    votantes: int32;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_candidato(var registro\_candidato: t\_registro\_candidato);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_candidato.localidad:=localidad\_salida

  else

    registro\_candidato.localidad:=random\_string(5+random(6));

  if (registro\_candidato.localidad<>localidad\_salida) then

  begin

    registro\_candidato.apellido:=random\_string(5+random(6));

    registro\_candidato.votos:=10000+random(high(int16)-10000);

    registro\_candidato.votantes:=100000+random(100001);

  end;

end;

procedure actualizar\_maximo\_cantidad(votos: int16; intendente: string; var votos\_cantidad: int16; var intendente\_cantidad: string);

begin

  if (votos>votos\_cantidad) then

  begin

    votos\_cantidad:=votos;

    intendente\_cantidad:=intendente;

  end;

end;

procedure actualizar\_maximo\_porcentaje(porcentaje: real; intendente: string; var votos\_porcentaje: real; var intendente\_porcentaje: string);

begin

  if (porcentaje>votos\_porcentaje) then

  begin

    votos\_porcentaje:=porcentaje;

    intendente\_porcentaje:=intendente;

  end;

end;

procedure leer\_candidatos(var intendente\_cantidad, intendente\_porcentaje: string);

var

  registro\_candidato: t\_registro\_candidato;

  votos\_cantidad: int16;

  porcentaje, votos\_porcentaje: real;

begin

  votos\_cantidad:=low(int16);

  votos\_porcentaje:=-9999999;

  leer\_candidato(registro\_candidato);

  while (registro\_candidato.localidad<>localidad\_salida) do

  begin

    actualizar\_maximo\_cantidad(registro\_candidato.votos,registro\_candidato.apellido,votos\_cantidad,intendente\_cantidad);

    porcentaje:=registro\_candidato.votos/registro\_candidato.votantes\*100;

    actualizar\_maximo\_porcentaje(porcentaje,registro\_candidato.apellido,votos\_porcentaje,intendente\_porcentaje);

    leer\_candidato(registro\_candidato);

  end;

end;

var

  intendente\_cantidad, intendente\_porcentaje: string;

begin

  randomize;

  intendente\_cantidad:='';

  intendente\_porcentaje:='';

  leer\_candidatos(intendente\_cantidad,intendente\_porcentaje);

  textcolor(green); write('El intendente que obtuvo la mayor cantidad de votos en la elección es '); textcolor(red); writeln(intendente\_cantidad);

  textcolor(green); write('El intendente que obtuvo el mayor porcentaje de votos en la elección es '); textcolor(red); write(intendente\_porcentaje);

end.

**Ejercicio 10.**

*Un centro de investigación de la UNLP está organizando la información de las 320 especies de plantas con las que trabajan. Para cada especie, se ingresa su nombre científico, tiempo promedio de vida (en meses), tipo de planta (por ej., árbol, conífera, arbusto, helecho, musgo, etc.), clima (templado, continental, subtropical, desértico, etc.) y países en el mundo donde se las encuentra. La información de las plantas se ingresa ordenada por tipo de planta y, para cada planta, la lectura de países donde se las encuentra finaliza al ingresar el país “zzz”. Al finalizar la lectura, informar:*

* *El tipo de planta con menor cantidad de plantas.*
* *El tiempo promedio de vida de las plantas de cada tipo.*
* *El nombre científico de las dos plantas más longevas.*
* *Los nombres de las plantas nativas de Argentina que se encuentran en regiones con clima subtropical.*
* *El nombre de la planta que se encuentra en más países.*

program TP3\_E10;

*{$codepage UTF8}*

uses crt;

const

  plantas\_total=320;

  pais\_salida='zzz';

  pais\_corte='Argentina'; clima\_corte='subtropical';

type

  t\_registro\_planta=record

    nombre: string;

    vida: int16;

    tipo: string;

    clima: string;

    pais: string;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_planta(var registro\_planta: t\_registro\_planta; planta, tipo: string; var tipo\_pos: int8);

var

  vector\_tipos: array[1..5] of string=('arbol', 'conifera', 'arbusto', 'helecho', 'musgo');

  vector\_climas: array[1..4] of string=('templado', 'continental', 'subtropical', 'desertico');

  i: int8;

begin

  i:=random(10);

  if (i=0) then

  begin

    registro\_planta.nombre:='Especie '+random\_string(5+random(6));

    registro\_planta.pais:=pais\_salida;

  end

  else

  begin

    registro\_planta.nombre:=planta;

    if (i<=5) then

      registro\_planta.pais:=pais\_corte

    else

      registro\_planta.pais:=random\_string(5+random(6));

  end;

  i:=random(20);

  if (i=0) then

  begin

    registro\_planta.nombre:='Especie '+random\_string(5+random(6));

    tipo\_pos:=tipo\_pos+1;

    if (tipo\_pos<=5) then

      registro\_planta.tipo:=vector\_tipos[tipo\_pos]

    else

      registro\_planta.tipo:='Tipo '+random\_string(5+random(6));

  end

  else

    if (tipo\_pos<=5) then

      registro\_planta.tipo:=vector\_tipos[tipo\_pos]

    else

      registro\_planta.tipo:=tipo;

  registro\_planta.vida:=1+random(100);

  registro\_planta.clima:=vector\_climas[1+random(4)];

end;

procedure actualizar\_minimo(plantas\_tipo: int16; tipo: string; var plantas\_min: int16; var tipo\_min: string);

begin

  if (plantas\_tipo<plantas\_min) then

  begin

    plantas\_min:=plantas\_tipo;

    tipo\_min:=tipo;

  end;

end;

function calcular\_tiempo\_promedio(vida\_tipo, plantas\_tipo\_paises: int16): real;

begin

  calcular\_tiempo\_promedio:=vida\_tipo/plantas\_tipo\_paises;

end;

procedure actualizar\_maximos(vida: int16; planta: string; var vida\_max1, vida\_max2: int16; var planta\_max1, planta\_max2: string);

begin

  if (vida>vida\_max1) then

  begin

    vida\_max2:=vida\_max1;

    planta\_max2:=planta\_max1;

    vida\_max1:=vida;

    planta\_max1:=planta;

  end

  else

    if (vida>vida\_max2) then

    begin

      vida\_max2:=vida;

      planta\_max2:=planta;

    end;

end;

procedure actualizar\_maximo(paises\_planta: int16; planta: string; var paises\_max3: int16; var planta\_max3: string);

begin

  if (paises\_planta>paises\_max3) then

  begin

    paises\_max3:=paises\_planta;

    planta\_max3:=planta;

  end;

end;

procedure leer\_plantas(var tipo\_min, planta\_max1, planta\_max2, planta\_max3: string);

var

  registro\_planta: t\_registro\_planta;

  tipo\_pos: int8;

  plantas, plantas\_tipo, plantas\_min, vida\_tipo, plantas\_tipo\_paises, vida\_max1, vida\_max2, paises\_planta, paises\_max3: int16;

  tipo, planta: string;

begin

  plantas:=0;

  plantas\_min:=high(int16);

  vida\_max1:=low(int16); vida\_max2:=low(int16);

  paises\_max3:=low(int16);

  planta:='Especie XXX'; tipo:=''; tipo\_pos:=1;

  leer\_planta(registro\_planta,planta,tipo,tipo\_pos);

  while (plantas<plantas\_total) do

  begin

    tipo:=registro\_planta.tipo;

    plantas\_tipo:=0;

    vida\_tipo:=0; plantas\_tipo\_paises:=0;

    while ((plantas<plantas\_total) and (registro\_planta.tipo=tipo)) do

    begin

      planta:=registro\_planta.nombre;

      plantas\_tipo:=plantas\_tipo+1;

      paises\_planta:=0;

      while ((registro\_planta.tipo=tipo) and (registro\_planta.pais<>pais\_salida)) do

      begin

        vida\_tipo:=vida\_tipo+registro\_planta.vida;

        plantas\_tipo\_paises:=plantas\_tipo\_paises+1;

        actualizar\_maximos(registro\_planta.vida,registro\_planta.nombre,vida\_max1,vida\_max2,planta\_max1,planta\_max2);

        if ((registro\_planta.pais=pais\_corte) and (registro\_planta.clima=clima\_corte)) then

        begin

          textcolor(green); write('El nombre de la planta nativa de '); textcolor(yellow); write(pais\_corte); textcolor(green); write(' que se encuentran en una región con clima '); textcolor(yellow); write(clima\_corte); textcolor(green); write(' es '); textcolor(red); writeln(planta);

        end;

        paises\_planta:=paises\_planta+1;

        leer\_planta(registro\_planta,planta,tipo,tipo\_pos);

      end;

      if (registro\_planta.tipo=tipo) then

        registro\_planta.pais:=random\_string(5+random(6));

      actualizar\_maximo(paises\_planta,planta,paises\_max3,planta\_max3);

      plantas:=plantas+1;

    end;

    actualizar\_minimo(plantas\_tipo,tipo,plantas\_min,tipo\_min);

    textcolor(green); write('El tiempo promedio de vida (en meses) de las plantas de tipo '); textcolor(red); write(tipo); textcolor(green); write(' es '); textcolor(red); writeln(calcular\_tiempo\_promedio(vida\_tipo,plantas\_tipo\_paises):0:2);

  end;

end;

var

  tipo\_min, planta\_max1, planta\_max2, planta\_max3: string;

begin

  randomize;

  tipo\_min:='';

  planta\_max1:=''; planta\_max2:='';

  planta\_max3:='';

  leer\_plantas(tipo\_min,planta\_max1,planta\_max2,planta\_max3);

  textcolor(green); write('El tipo de planta con menor cantidad de plantas es '); textcolor(red); writeln(tipo\_min);

  textcolor(green); write('Los nombres científicos de las dos plantas más longevas son '); textcolor(red); write(planta\_max1); textcolor(green); write(' y '); textcolor(red); writeln(planta\_max2);

  textcolor(green); write('El nombre de la planta que se encuentra en más países es '); textcolor(red); write(planta\_max3);

end.

**Ejercicio 11.**

*Una compañía de vuelos internacionales está analizando la información de todos los vuelos realizados por sus aviones durante todo el año 2019. De cada vuelo, se conoce el código de avión, país de salida, país de llegada, cantidad de kilómetros recorridos y porcentaje de ocupación del avión. La información se ingresa ordenada por código de avión y, para cada avión, por país de salida. La lectura finaliza al ingresar el código 44. Informar:*

* *Los dos aviones que más kilómetros recorrieron y los dos aviones que menos kilómetros recorrieron.*
* *El avión que salió desde más países diferentes.*
* *La cantidad de vuelos de más de 5.000 km que no alcanzaron el 60% de ocupación del avión.*
* *La cantidad de vuelos de menos de 10.000 km que llegaron a Australia o a Nueva Zelanda.*

program TP3\_E11;

*{$codepage UTF8}*

uses crt;

const

  avion\_salida=44;

  kms\_corte1=5000.0; ocupacion\_corte=60.0;

  kms\_corte2=10000.0; pais\_corte1='Australia'; pais\_corte2='Nueva Zelanda';

type

  t\_registro\_vuelo=record

    avion: int16;

    pais\_salida: string;

    pais\_llegada: string;

    kms: real;

    ocupacion: real;

  end;

function random\_string(length: int8): string;

var

  i: int8;

  string\_aux: string;

begin

  string\_aux:='';

  for i:= 1 to length do

    string\_aux:=string\_aux+chr(ord('A')+random(26));

  random\_string:=string\_aux;

end;

procedure leer\_vuelo(var registro\_vuelo: t\_registro\_vuelo; avion: int16; pais\_salida: string);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_vuelo.avion:=avion\_salida

  else if (i<=50) then

    registro\_vuelo.avion:=avion

  else

    registro\_vuelo.avion:=1+random(high(int16));

  if (registro\_vuelo.avion<>avion\_salida) then

  begin

    i:=random(2);

    if (i=0) then

      registro\_vuelo.pais\_salida:=pais\_salida

    else

      registro\_vuelo.pais\_salida:=random\_string(5+random(6));

    i:=1+random(3);

    if (i=1) then

      registro\_vuelo.pais\_llegada:=pais\_corte1

    else if (i=2) then

      registro\_vuelo.pais\_llegada:=pais\_corte2

    else

      registro\_vuelo.pais\_llegada:=random\_string(random(10));

    registro\_vuelo.kms:=100+random(9901);

    registro\_vuelo.ocupacion:=1+random(991)/10;

  end;

end;

procedure actualizar\_maximos(kms\_avion: real; avion: int16; var kms\_max1, kms\_max2: real; var avion\_max1, avion\_max2: int16);

begin

  if (kms\_avion>kms\_max1) then

  begin

    kms\_max2:=kms\_max1;

    avion\_max2:=avion\_max1;

    kms\_max1:=kms\_avion;

    avion\_max1:=avion;

  end

  else

    if (kms\_avion>kms\_max2) then

    begin

      kms\_max2:=kms\_avion;

      avion\_max2:=avion;

    end;

end;

procedure actualizar\_minimos(var kms\_avion: real; avion: int16; var kms\_min1, kms\_min2: real; var avion\_min1, avion\_min2: int16);

begin

  if (kms\_avion<kms\_min1) then

  begin

    kms\_min2:=kms\_min1;

    avion\_min2:=avion\_min1;

    kms\_min1:=kms\_avion;

    avion\_min1:=avion;

  end

  else

    if (kms\_avion<kms\_min2) then

    begin

      kms\_min2:=kms\_avion;

      avion\_min2:=avion;

    end;

end;

procedure actualizar\_maximo(paises\_avion, avion: int16; var paises\_max3, avion\_max3: int16);

begin

  if (paises\_avion>paises\_max3) then

  begin

    paises\_max3:=paises\_avion;

    avion\_max3:=avion;

  end;

end;

procedure leer\_vuelos(var avion\_max1, avion\_max2, avion\_min1, avion\_min2, avion\_max3, vuelos\_corte1, vuelos\_corte2: int16);

var

  registro\_vuelo: t\_registro\_vuelo;

  avion, paises\_avion, paises\_max3: int16;

  kms\_avion, kms\_max1, kms\_max2, kms\_min1, kms\_min2: real;

  pais: string;

begin

  kms\_max1:=-9999999; kms\_max2:=-9999999;

  kms\_min1:=9999999; kms\_min2:=9999999;

  paises\_max3:=low(int16);

  avion:=1; pais:='XXX';

  leer\_vuelo(registro\_vuelo,avion,pais);

  while (registro\_vuelo.avion<>avion\_salida) do

  begin

    avion:=registro\_vuelo.avion;

    kms\_avion:=0;

    paises\_avion:=0;

    while ((registro\_vuelo.avion<>avion\_salida) and (registro\_vuelo.avion=avion)) do

    begin

      pais:=registro\_vuelo.pais\_salida;

      paises\_avion:=paises\_avion+1;

      while ((registro\_vuelo.avion<>avion\_salida) and (registro\_vuelo.avion=avion) and (registro\_vuelo.pais\_salida=pais)) do

      begin

        kms\_avion:=kms\_avion+registro\_vuelo.kms;

        if ((registro\_vuelo.kms>kms\_corte1) and (registro\_vuelo.ocupacion<ocupacion\_corte)) then

          vuelos\_corte1:=vuelos\_corte1+1;

        if ((registro\_vuelo.kms<kms\_corte2) and ((registro\_vuelo.pais\_llegada=pais\_corte1) or (registro\_vuelo.pais\_llegada=pais\_corte2))) then

          vuelos\_corte2:=vuelos\_corte2+1;

        leer\_vuelo(registro\_vuelo,avion,pais);

      end;

    end;

    actualizar\_maximos(kms\_avion,avion,kms\_max1,kms\_max2,avion\_max1,avion\_max2);

    actualizar\_minimos(kms\_avion,avion,kms\_min1,kms\_min2,avion\_min1,avion\_min2);

    actualizar\_maximo(paises\_avion,avion,paises\_max3,avion\_max3);

  end;

end;

var

  avion\_max1, avion\_max2, avion\_min1, avion\_min2, avion\_max3, vuelos\_corte1, vuelos\_corte2: int16;

begin

  randomize;

  avion\_max1:=0; avion\_max2:=0; avion\_min1:=0; avion\_min2:=0;

  avion\_max3:=0;

  vuelos\_corte1:=0;

  vuelos\_corte2:=0;

  leer\_vuelos(avion\_max1,avion\_max2,avion\_min1,avion\_min2,avion\_max3,vuelos\_corte1,vuelos\_corte2);

  textcolor(green); write('Los dos aviones que más kilómetros recorrieron son '); textcolor(red); write(avion\_max1); textcolor(green); write(' y '); textcolor(red); writeln(avion\_max2);

  textcolor(green); write('Los dos aviones que menos kilómetros recorrieron son '); textcolor(red); write(avion\_min1); textcolor(green); write(' y '); textcolor(red); writeln(avion\_min2);

  textcolor(green); write('El avión que salió de más países diferentes es '); textcolor(red); writeln(avion\_max3);

  textcolor(green); write('La cantidad de vuelos de más de '); textcolor(yellow); write(kms\_corte1:0:2); textcolor(green); write(' kms que no alcanzaron el '); textcolor(yellow); write(ocupacion\_corte:0:2); textcolor(green); write('% de ocupación del avión es '); textcolor(red); writeln(vuelos\_corte1);

  textcolor(green); write('La cantidad de vuelos de menos de '); textcolor(yellow); write(kms\_corte2:0:2); textcolor(green); write(' kms que llegaron a '); textcolor(yellow); write(pais\_corte1); textcolor(green); write(' o a '); textcolor(yellow); write(pais\_corte2); textcolor(green); write(' es '); textcolor(red); write(vuelos\_corte2);

end.

**Ejercicio 12.**

*En la “Práctica 1 - Ejercicios Adicionales”, se resolvieron 3 problemas complejos sin utilizar módulos. Al carecer de herramientas para modularizar, esos programas resultaban difíciles de leer, de extender y de depurar. En la “Práctica 2 (parte 2) - Ejercicios Adicionales”, se adaptaron los 3 problemas para utilizar módulos y, así, organizar mejor el programa. Ahora, se pueden incluir los registros y, así, seguir mejorando los programas. Para cada caso, analizar:*

* *¿Qué entidades del programa conviene representar como registros?*
* *¿Qué atributos de cada entidad deben incluirse en los registros?*
* *¿Qué cambios deben realizarse en los módulos implementados en la Práctica 2 para aprovechar los nuevos tipos de datos? ¿Conviene seguir utilizando los mismos módulos en todos los casos?*

*Una vez realizado el análisis, modificar los 3 problemas, utilizando registros para representar los datos del programa. Al finalizar cada problema, comparar la solución usando registros y módulos con la solución sin registros y con módulos (Práctica 2) y con la solución sin registros ni módulos (Práctica 1).*

* *¿Qué diferencias se observan?*
* *¿Qué similitudes se encuentran?*

Ejercicio 1:

program TP3\_E12a;

*{$codepage UTF8}*

uses crt;

const

  empresa\_salida=100;

  monto\_corte=50000.0;

type

  t\_registro\_empresa=record

    empresa: int16;

    inversiones: int16;

    monto\_total: real;

  end;

procedure leer\_inversiones(empresa, inversiones: int16; var monto\_total: real);

var

  i: int16;

  monto: real;

begin

  monto\_total:=0;

  for i:= 1 to inversiones do

  begin

    monto:=1+random(1000);

    monto\_total:=monto\_total+monto;

  end;

end;

procedure leer\_empresa(var registro\_empresa: t\_registro\_empresa);

var

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_empresa.empresa:=empresa\_salida

  else

    registro\_empresa.empresa:=1+random(high(int16));

  registro\_empresa.inversiones:=1+random(1000);

  leer\_inversiones(registro\_empresa.empresa,registro\_empresa.inversiones,registro\_empresa.monto\_total);

end;

procedure calcular\_a(empresa, inversiones: int16; monto\_total: real);

begin

  textcolor(green); write('El monto promedio de las inversiones de la empresa '); textcolor(yellow); write(empresa); textcolor(green); write(' es '); textcolor(red); writeln(monto\_total/inversiones:0:2);

end;

procedure calcular\_b(monto\_total: real; empresa: int16; var monto\_max: real; var empresa\_max: int16);

begin

  if (monto\_total>monto\_max) then

  begin

    monto\_max:=monto\_total;

    empresa\_max:=empresa;

  end;

end;

procedure calcular\_c(monto\_total: real; var empresas\_corte: int16);

begin

  if (monto\_total>monto\_corte) then

    empresas\_corte:=empresas\_corte+1;

end;

procedure leer\_empresas(var empresa\_max, empresas\_corte: int16);

var

  registro\_empresa: t\_registro\_empresa;

  monto\_max: real;

begin

  monto\_max:=-9999999;

  repeat

    leer\_empresa(registro\_empresa);

    calcular\_a(registro\_empresa.empresa,registro\_empresa.inversiones,registro\_empresa.monto\_total);

    calcular\_b(registro\_empresa.monto\_total,registro\_empresa.empresa,monto\_max,empresa\_max);

    calcular\_c(registro\_empresa.monto\_total,empresas\_corte);

  until (registro\_empresa.empresa=empresa\_salida);

end;

var

  empresa\_max, empresas\_corte: int16;

begin

  randomize;

  empresa\_max:=0;

  empresas\_corte:=0;

  leer\_empresas(empresa\_max,empresas\_corte);

  textcolor(green); write('El código de la empresa con mayor monto total invertido es '); textcolor(red); writeln(empresa\_max);

  textcolor(green); write('La cantidad de empresas con inversiones de más de $'); textcolor(yellow); write(monto\_corte:0:2); textcolor(green); write(' es '); textcolor(red); write(empresas\_corte);

end.

Ejercicio 2:

program TP3\_E12b;

*{$codepage UTF8}*

uses crt;

const

  condicion\_i='I'; condicion\_r='R';

  autoeva\_total=5;

  nota\_incumple=-1;

  legajo\_salida=-1;

  nota\_corte=4;

  promedio\_corte=6.5;

  nota\_cero=0;

  nota\_diez=10;

  presente\_corte=0.75;

type

  t\_registro\_alumno=record

    legajo: int16;

    condicion: char;

    presente: int8;

    nota\_total: int8;

    notas\_cero: int8;

    notas\_diez: int8;

  end;

procedure leer\_notas(var presente, nota\_total, notas\_cero, notas\_diez: int8);

var

  i, nota: int8;

begin

  presente:=0; nota\_total:=0; notas\_cero:=0; notas\_diez:=0;

  for i:= 1 to autoeva\_total do

  begin

    nota:=nota\_incumple+random(12);

    if ((nota<>nota\_incumple) and (nota>=nota\_corte)) then

      presente:=presente+1;

    if (nota<>nota\_incumple) then

      nota\_total:=nota\_total+nota;

    if (nota=nota\_cero) then

      notas\_cero:=notas\_cero+1;

    if (nota=nota\_diez) then

      notas\_diez:=notas\_diez+1;

  end;

end;

procedure leer\_alumno(var registro\_alumno: t\_registro\_alumno);

var

  vector\_condiciones: array[1..2] of char=(condicion\_i, condicion\_r);

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_alumno.legajo:=legajo\_salida

  else

    registro\_alumno.legajo:=1+random(high(int16));

  if (registro\_alumno.legajo<>legajo\_salida) then

  begin

    registro\_alumno.condicion:=vector\_condiciones[1+random(2)];

    leer\_notas(registro\_alumno.presente,registro\_alumno.nota\_total,registro\_alumno.notas\_cero,registro\_alumno.notas\_diez);

  end;

end;

procedure calcular\_ab(condicion: char; presente: int8; var ingresantes\_total, ingresantes\_parcial, recursantes\_total, recursantes\_parcial: int16);

begin

  if (condicion=condicion\_i) then

  begin

    if (presente>=presente\_corte\*autoeva\_total) then

      ingresantes\_parcial:=ingresantes\_parcial+1;

    ingresantes\_total:=ingresantes\_total+1;

  end

  else

  begin

    if (presente>=presente\_corte\*autoeva\_total) then

      recursantes\_parcial:=recursantes\_parcial+1;

    recursantes\_total:=recursantes\_total+1;

  end;

end;

procedure calcular\_c(presente: int8; var alumnos\_autoeva: int16);

begin

  if (presente=autoeva\_total) then

    alumnos\_autoeva:=alumnos\_autoeva+1;

end;

procedure calcular\_d(nota\_total: int8; var alumnos\_corte: int16);

begin

  if (nota\_total/autoeva\_total>promedio\_corte) then

    alumnos\_corte:=alumnos\_corte+1;

end;

procedure calcular\_e(notas\_cero: int8; var alumnos\_cero: int16);

begin

  if (notas\_cero>=1) then

    alumnos\_cero:=alumnos\_cero+1;

end;

procedure calcular\_f(notas\_diez: int8; legajo: int16; var notas\_diez\_max1, notas\_diez\_max2: int8; var legajo\_diez\_max1, legajo\_diez\_max2: int16);

begin

  if (notas\_diez>notas\_diez\_max1) then

  begin

    notas\_diez\_max2:=notas\_diez\_max1;

    legajo\_diez\_max2:=legajo\_diez\_max1;

    notas\_diez\_max1:=notas\_diez;

    legajo\_diez\_max1:=legajo;

  end

  else

    if (notas\_diez>notas\_diez\_max2) then

    begin

      notas\_diez\_max2:=notas\_diez;

      legajo\_diez\_max2:=legajo;

    end;

end;

procedure calcular\_g(notas\_cero: int8; legajo: int16; var notas\_cero\_max1, notas\_cero\_max2: int8; var legajo\_cero\_max1, legajo\_cero\_max2: int16);

begin

  if (notas\_cero>notas\_cero\_max1) then

  begin

    notas\_cero\_max2:=notas\_cero\_max1;

    legajo\_cero\_max2:=legajo\_cero\_max1;

    notas\_cero\_max1:=notas\_cero;

    legajo\_cero\_max1:=legajo;

  end

  else

    if (notas\_cero>notas\_cero\_max2) then

    begin

      notas\_cero\_max2:=notas\_cero;

      legajo\_cero\_max2:=legajo;

    end;

end;

procedure leer\_alumnos(var ingresantes\_parcial, ingresantes\_total, recursantes\_parcial, recursantes\_total, alumnos\_autoeva, alumnos\_corte, alumnos\_cero, legajo\_diez\_max1, legajo\_diez\_max2, legajo\_cero\_max1, legajo\_cero\_max2: int16);

var

  registro\_alumno: t\_registro\_alumno;

  notas\_diez\_max1, notas\_diez\_max2, notas\_cero\_max1, notas\_cero\_max2: int8;

begin

  notas\_diez\_max1:=0; notas\_diez\_max2:=0;

  notas\_cero\_max1:=0; notas\_cero\_max2:=0;

  leer\_alumno(registro\_alumno);

  while (registro\_alumno.legajo<>legajo\_salida) do

  begin

    calcular\_ab(registro\_alumno.condicion,registro\_alumno.presente,ingresantes\_total,ingresantes\_parcial,recursantes\_total,recursantes\_parcial);

    calcular\_c(registro\_alumno.presente,alumnos\_autoeva);

    calcular\_d(registro\_alumno.nota\_total,alumnos\_corte);

    calcular\_e(registro\_alumno.notas\_cero,alumnos\_cero);

    calcular\_f(registro\_alumno.notas\_diez,registro\_alumno.legajo,notas\_diez\_max1,notas\_diez\_max2,legajo\_diez\_max1,legajo\_diez\_max2);

    calcular\_g(registro\_alumno.notas\_cero,registro\_alumno.legajo,notas\_cero\_max1,notas\_cero\_max2,legajo\_cero\_max1,legajo\_cero\_max2);

    leer\_alumno(registro\_alumno);

  end;

end;

var

  ingresantes\_parcial, ingresantes\_total, recursantes\_parcial, recursantes\_total, alumnos\_autoeva, alumnos\_corte, alumnos\_cero, legajo\_diez\_max1, legajo\_diez\_max2, legajo\_cero\_max1, legajo\_cero\_max2: int16;

begin

randomize;

  ingresantes\_parcial:=0; ingresantes\_total:=0;

  recursantes\_parcial:=0; recursantes\_total:=0;

  alumnos\_autoeva:=0;

  alumnos\_corte:=0;

  alumnos\_cero:=0;

  legajo\_diez\_max1:=0; legajo\_diez\_max2:=0;

  legajo\_cero\_max1:=0; legajo\_cero\_max2:=0;

  leer\_alumnos(ingresantes\_parcial,ingresantes\_total,recursantes\_parcial,recursantes\_total,alumnos\_autoeva,alumnos\_corte,alumnos\_cero,legajo\_diez\_max1,legajo\_diez\_max2,legajo\_cero\_max1,legajo\_cero\_max2);

  if ((ingresantes\_total>0) or (recursantes\_total>0)) then

  begin

    if (ingresantes\_total>0) then

    begin

      textcolor(green); write('La cantidad de alumnos INGRESANTES en condiciones de rendir el parcial y el porcentaje sobre el total de alumnos INGRESANTES son '); textcolor(red); write(ingresantes\_parcial); textcolor(green); write(' y '); textcolor(red); write(ingresantes\_parcial/ingresantes\_total\*100:0:2); textcolor(green); writeln('%, respectivamente');

    end

    else

    begin

      textcolor(red); writeln('No hay alumnos INGRESANTES (I)');

    end;

    if (recursantes\_total>0) then

    begin

      textcolor(green); write('La cantidad de alumnos RECURSANTES en condiciones de rendir el parcial y el porcentaje sobre el total de alumnos RECURSANTES son '); textcolor(red); write(recursantes\_parcial); textcolor(green); write(' y '); textcolor(red); write(recursantes\_parcial/recursantes\_total\*100:0:2); textcolor(green); writeln('%, respectivamente');

    end

    else

    begin

      textcolor(red); writeln('No hay alumnos RECURSANTES (R)');

    end;

    textcolor(green); write('La cantidad de alumnos que aprobaron todas las autoevaluaciones es '); textcolor(red); writeln(alumnos\_autoeva);

    textcolor(green); write('La cantidad de alumnos cuya nota promedio fue mayor a '); textcolor(yellow); write(promedio\_corte:0:2); textcolor(green); write(' puntos es '); textcolor(red); writeln(alumnos\_corte);

    textcolor(green); write('La cantidad de alumnos que obtuvieron cero puntos en, al menos, una autoevaluación es '); textcolor(red); writeln(alumnos\_cero);

    textcolor(green); write('Los legajos de los dos alumnos con mayor cantidad de autoevaluaciones con nota 10 (diez) son '); textcolor(red); write(legajo\_diez\_max1); textcolor(green); write(' y '); textcolor(red); writeln(legajo\_diez\_max2);

    textcolor(green); write('Los legajos de los dos alumnos con mayor cantidad de autoevaluaciones con nota 0 (cero) son '); textcolor(red); write(legajo\_cero\_max1); textcolor(green); write(' y '); textcolor(red); write(legajo\_cero\_max2);

  end

  else

  begin

    textcolor(red); write('No hay alumnos INGRESANTES (I) o RECURSANTES (R)');

  end;

end.

Ejercicio 3:

program TP3\_E12c;

*{$codepage UTF8}*

uses crt;

const

  tanque\_r='R'; tanque\_c='C';

  tanque\_salida='Z';

  alto\_corte=1.40;

  volumen\_corte=800.0;

type

  t\_registro\_tanque=record

    tanque: char;

    ancho: real;

    largo: real;

    alto: real;

    radio: real;

    volumen: real;

  end;

procedure leer\_tanque(var registro\_tanque: t\_registro\_tanque);

var

  vector\_tanques: array[1..2] of char=(tanque\_r, tanque\_c);

  i: int8;

begin

  i:=random(100);

  if (i=0) then

    registro\_tanque.tanque:=tanque\_salida

  else

    registro\_tanque.tanque:=vector\_tanques[1+random(2)];

  if (registro\_tanque.tanque<>tanque\_salida) then

  begin

    if (registro\_tanque.tanque=tanque\_r) then

    begin

      registro\_tanque.ancho:=1+random(391)/10;

      registro\_tanque.largo:=1+random(391)/10;

      registro\_tanque.alto:=1+random(21)/10;

      registro\_tanque.volumen:=registro\_tanque.ancho\*registro\_tanque.largo\*registro\_tanque.alto;

      registro\_tanque.radio:=-1;

    end

    else

    begin

      registro\_tanque.radio:=1+random(391)/10;

      registro\_tanque.alto:=1+random(21)/10;

      registro\_tanque.volumen:=pi\*registro\_tanque.radio\*registro\_tanque.radio\*registro\_tanque.alto;

      registro\_tanque.ancho:=-1;

      registro\_tanque.largo:=-1;

    end;

  end;

end;

procedure calcular\_a(volumen: real; var volumen\_max1, volumen\_max2: real);

begin

  if (volumen>volumen\_max1) then

  begin

    volumen\_max2:=volumen\_max1;

    volumen\_max1:=volumen;

  end

  else

    if (volumen>volumen\_max2) then

      volumen\_max2:=volumen;

end;

procedure calcular\_bc(tanque: char; volumen: real; var volumen\_total\_c, volumen\_total\_r: real; var tanques\_c, tanques\_r: int16);

begin

  if (tanque=tanque\_c) then

  begin

    volumen\_total\_c:=volumen\_total\_c+volumen;

    tanques\_c:=tanques\_c+1;

  end

  else

  begin

    volumen\_total\_r:=volumen\_total\_r+volumen;

    tanques\_r:=tanques\_r+1;

  end;

end;

procedure calcular\_d(alto: real; var tanques\_corte\_alto: int16);

begin

  if (alto<alto\_corte) then

    tanques\_corte\_alto:=tanques\_corte\_alto+1;

end;

procedure calcular\_e(volumen: real; var tanques\_corte\_volumen: int16);

begin

  if (volumen<volumen\_corte) then

    tanques\_corte\_volumen:=tanques\_corte\_volumen+1;

end;

procedure leer\_tanques(var volumen\_max1, volumen\_max2, volumen\_total\_c, volumen\_total\_r: real; var tanques\_c, tanques\_r, tanques\_corte\_alto, tanques\_corte\_volumen: int16);

var

  registro\_tanque: t\_registro\_tanque;

begin

  leer\_tanque(registro\_tanque);

  while (registro\_tanque.tanque<>tanque\_salida) do

  begin

    calcular\_a(registro\_tanque.volumen,volumen\_max1,volumen\_max2);

    calcular\_bc(registro\_tanque.tanque,registro\_tanque.volumen,volumen\_total\_c,volumen\_total\_r,tanques\_c,tanques\_r);

    calcular\_d(registro\_tanque.alto,tanques\_corte\_alto);

    calcular\_e(registro\_tanque.volumen,tanques\_corte\_volumen);

    leer\_tanque(registro\_tanque);

  end;

end;

var

  tanques\_c, tanques\_r, tanques\_corte\_alto, tanques\_corte\_volumen: int16;

  volumen\_max1, volumen\_max2, volumen\_total\_c, volumen\_total\_r: real;

begin

  randomize;

  volumen\_max1:=0; volumen\_max2:=0;

  tanques\_c:=0; volumen\_total\_c:=0;

  tanques\_r:=0; volumen\_total\_r:=0;

  tanques\_corte\_alto:=0;

  tanques\_corte\_volumen:=0;

  leer\_tanques(volumen\_max1,volumen\_max2,volumen\_total\_c,volumen\_total\_r,tanques\_c,tanques\_r,tanques\_corte\_alto,tanques\_corte\_volumen);

  if ((tanques\_c>0) or (tanques\_r>0)) then

  begin

    textcolor(green); write('El volumen de los mayores tanques vendidos es '); textcolor(red); write(volumen\_max1:0:2); textcolor(green); write(' y '); textcolor(red); writeln(volumen\_max2:0:2);

    if (tanques\_c>0) then

    begin

      textcolor(green); write('El volumen promedio de todos los tanques cilíndricos (C) vendidos es '); textcolor(red); writeln(volumen\_total\_c/tanques\_c:0:2);

    end

    else

    begin

      textcolor(red); writeln('No hay tanques cilíndricos (C) vendidos');

    end;

    if (tanques\_r>0) then

    begin

      textcolor(green); write('El volumen promedio de todos los tanques rectangulares (R) vendidos es '); textcolor(red); writeln(volumen\_total\_r/tanques\_r:0:2);

    end

    else

    begin

      textcolor(red); writeln('No hay tanques rectangulares (R) vendidos');

    end;

    textcolor(green); write('La cantidad de tanques cuyo alto es menor a '); textcolor(yellow); write(alto\_corte:0:2); textcolor(green); write(' metros es '); textcolor(red); writeln(tanques\_corte\_alto);

    textcolor(green); write('La cantidad de tanques cuyo volumen es menor a '); textcolor(yellow); write(volumen\_corte:0:2); textcolor(green); write(' metros cúbicos es '); textcolor(red); write(tanques\_corte\_volumen);

  end

  else

  begin

    textcolor(red); write('No hay tanques cilíndricos (C) o rectangulares (R) vendidos');

  end;

end.