

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;
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 leidos 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;
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;
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));
    end;
    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;
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;
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;
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;
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.
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(41)/10;
            registro_procesador.transistores:=vector_nms[1+random(4)];
        end;
    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;
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 linea 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;
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;
        end;
    if (becarios<becarios_min2) then
        begin
            becarios_min1:=becarios;
            centro_min1:=centro;
        end;
end;

```

```

    becarios_min1:=becarios;
    centro_min1:=centro;
end
else
    if (becarios<becarios_min2) then
        begin
            becarios_min2:=becarios;
            centro_min2:=centro;
        end;
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,becario
s_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_ma
x);
        end;
    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));
    end;
    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;
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;
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;
    contar_pares_impares:=(pares=impares);
  end;
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,escuel
a_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_cant
idad,intendente_cantidad);
        porcentaje:=registro_candidato.votos/registro_candidato.votantes*100;
        actualizar_maximo_porcentaje(porcentaje,registro_candidato.apellido,votos_porcentaje,inten
dente_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(11);
  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
    tipo_pos:=tipo_pos+1;
    if ((tipo_pos>=1) and (tipo_pos<=5)) then
        registro_planta.tipo:=vector_tipos[tipo_pos]
    else
        registro_planta.tipo:='Tipo '+random_string(5+random(6));
    if (registro_planta.nombre=planta) then
        registro_planta.nombre:='Especie '+random_string(5+random(6))
    else
        registro_planta.pais:=random_string(5+random(6));
    end
else
    if ((tipo_pos>=1) and (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;
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 XXX'; 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;
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(101);
  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;
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;
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;
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;
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_cort
e2);
    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: 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;
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;
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,ingresant
es_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_ma
x2,legajo_diez_max1,legajo_diez_max2);

```

```

    calcular_g(registro_alumno.notas_cero,registro_alumno.legajo,notas_cero_max1,notas_cero_ma
x2,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,alu
mnos_autoeva,alumnos_corte,alumnos_cero,legajo_diez_max1,legajo_diez_max2,legajo_cero_max1,leg
ajo_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;
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;
    end;
    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;
  end;
  registro_tanque.ancho:=-1;
  registro_tanque.largo:=-1;
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;
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;
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,t
anques_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.
end.
```