

## **Trabajo Práctico N° 2.2:** **Funciones, Procedimientos y Parámetros.**

### **Ejercicio 1.**

*Responder las preguntas en relación al siguiente programa:*

```
program TP2_E1;
{$codepage UTF8}
uses crt;
procedure suma(num1: integer; var num2: integer);
begin
    num2:=num1+num2;
    num1:=0;
end;
var
    i, x: integer;
begin
    read(x);
    for i:= 1 to 5 do
        suma(i,x);
        write(x);
    end.
end.
```

(a) *¿Qué imprime si se lee el valor 10 en la variable x?*

```
program TP2_E1a;
{$codepage UTF8}
uses crt;
procedure suma(num1: integer; var num2: integer);
begin
    num2:=num1+num2;
    num1:=0;
end;
var
    i, x: integer;
begin
    read(x);
    for i:= 1 to 5 do
        suma(i,x);
        write(x);
    end.
end.
```

Si se lee el valor 10 en la variable “x”, se imprime el número 25.

(b) *¿Qué imprime si se lee el valor 10 en la variable x y se cambia el encabezado del procedure por: procedure suma(num1: integer; num2: integer);?*

```
program TP2_E1b;
{$codepage UTF8}
uses crt;
procedure suma(num1: integer; num2: integer);
begin
    num2:=num1+num2;
    num1:=0;
end;
```

```
var
  i, x: integer;
begin
  read(x);
  for i:= 1 to 5 do
    suma(i,x);
  write(x);
end.
```

Si se lee el valor 10 en la variable “x” y se cambia el encabezado del *procedure* por *procedure suma(num1: integer; num2:integer)*, se imprime el número 10.

(c) ¿Qué sucede si se cambia el encabezado del *procedure* por: *procedure suma(var num1: integer; var num2: integer);*?

```
program TP2_E1c;
{$codepage UTF8}
uses crt;
procedure suma(var num1: integer; var num2: integer);
begin
  num2:=num1+num2;
  num1:=0;
end;
var
  i, x: integer;
begin
  read(x);
  for i:= 1 to 5 do
    suma(i,x);
  write(x);
end.
```

Lo que sucede si se cambia el encabezado del *procedure* por *procedure suma(var num1: integer; var num2: integer)* es que se generará un error, ya que no es posible modificar el valor de una variable índice (en este caso, “i”).

## Ejercicio 2.

*Responder la pregunta en relación al siguiente programa:*

```

program TP2_E2;
{$codepage UTF8}
uses crt;
procedure digParesImpares(num: integer; var par, impar: integer);
var
  dig: integer;
begin
  while (num<>0) do
  begin
    dig:=num mod 10;
    if (dig mod 2=0) then
      par:=par+1
    else
      impar:=impar+1;
    num:=num div 10;
  end;
end;
var
  dato, par, impar, total, cant: integer;
begin
  par:=0;
  impar:=0;
  repeat
    read(dato);
    digParesImpares(dato,par,impar);
  until (dato=100);
  writeln('Pares: ',par,' e Impares: ',impar);
end.

```

*¿Qué imprime si se lee la secuencia de valores 250, 35, 100?*

```

program TP2_E2;
{$codepage UTF8}
uses crt;
procedure digParesImpares(num: integer; var par, impar: integer);
var
  dig: integer;
begin
  while (num<>0) do
  begin
    dig:=num mod 10;
    if (dig mod 2=0) then
      par:=par+1
    else
      impar:=impar+1;
    num:=num div 10;
  end;
end;
var
  vector_nums: array[1..3] of integer=(250, 35, 100);
  dato, par, impar, pos: integer;
begin
  par:=0;
  impar:=0;
  pos:=0;
  repeat
    pos:=pos+1;
    dato:=vector_nums[pos];

```

```
    digParesImpares(dato,par,impar);  
until (dato=100);  
    textcolor(green); write('Pares: '); textcolor(red); write(par); textcolor(green); write(' e  
Impares: '); textcolor(red); write(impar);  
end.
```

Si se lee la secuencia de valores 250, 35 y 100, el programa imprime “*Pares: 4 e Impares: 4*”.

### Ejercicio 3.

Encontrar los 6 errores que existen en el siguiente programa. Utilizar los comentarios entre llaves como guía, indicar en qué línea se encuentra cada error y en qué consiste.

Con errores:

```
program TP2_E3;
{$codepage UTF8}
uses crt;
{Suma los números entre a y b y retorna el resultado en c}
procedure sumar(a, b, c: integer)
var
    suma: integer;
begin
    for i:= a to b do
        suma:=suma+i;
        c:=c+suma;
    end;
var
    result: integer;
begin
    result:=0;
    readln(a); readln(b);
    sumar(a,b,0);
    write('La suma total es ',result);
    {Averigua si el resultado final estuvo entre 10 y 30}
    ok:=((result>=10) or (result<=30));
    if (not ok) then
        write('La suma no quedó entre 10 y 30');
    end.
end.
```

Sin errores:

```
program TP2_E3;
{$codepage UTF8}
uses crt;
procedure sumar(a, b: integer; var c: integer);
var
    i, suma: integer;
begin
    suma:=0;
    for i:= a to b do
        suma:=suma+i;
        c:=c+suma;
    end;
var
    result, a, b: integer;
    ok: boolean;
begin
    randomize;
    result:=0;
    a:=random(100); b:=a+random(100-a);
    sumar(a,b,result);
    textcolor(green); write('La suma total es '); textcolor(red); writeln(result);
    ok:=((result>=10) or (result<=30));
    if (not ok) then
        begin
            textcolor(green); write('La suma no quedó entre 10 y 30');
        end;
    end.
end.
```

Los 6 errores que existen en el programa son:

1. Línea 3: En el *procedure* “*sumar*”, falta “;” al final de la instrucción.
2. Línea 3: En el *procedure* “*sumar*”, el parámetro “*c*” debe ser por referencia.
3. Línea 5: En el *procedure* “*sumar*”, falta crear la variable local al proceso correspondiente al índice “*i*”.
4. Línea 7: En el *procedure* “*sumar*”, falta inicializar la variable “*suma*”.
5. Línea 13: En el programa principal, falta crear las variables locales al programa “*a*” y “*b*”, como *integer*, y “*ok*”, como *boolean*.
6. Línea 17: En el programa principal, en la invocación del *procedure* “*sumar*”, se debe pasar la variable local al programa “*result*” como parámetro por referencia.

## Ejercicio 4.

*El siguiente programa intenta resolver un enunciado. Sin embargo, el código posee 5 errores. Indicar en qué línea se encuentra cada error y en qué consiste el error.*

*Enunciado: Realizar un programa que lea datos de 130 programadores Java de una empresa. De cada programador, se lee el número de legajo y el salario actual. El programa debe imprimir el total del dinero destinado por mes al pago de salarios y el salario del empleado con mayor legajo.*

Con errores:

```
program TP2_E4;
{$codepage UTF8}
uses crt;
procedure leerDatos(var legajo: integer; salario: real);
begin
    writeln('Ingrese el número de legajo y el salario');
    read(legajo);
    read(salario);
end;
procedure actualizarMaximo(nuevoLegajo: integer; nuevoSalario: real; var maxLegajo: integer);
var
    maxSalario: real;
begin
    if (nuevoLegajo>maxLegajo) then
    begin
        maxLegajo:=nuevoLegajo;
        maxSalario:=nuevoSalario;
    end;
end;
var
    legajo, maxLegajo, i: integer;
    salario, maxSalario: real;
begin
    sumaSalarios:=0;
    for i:= 1 to 130 do
    begin
        leerDatos(salario,legajo);
        actualizarMaximo(legajo,salario,maxLegajo);
        sumaSalarios:=sumaSalarios+salario;
    end;
    writeln('En todo el mes se gastan ',sumaSalarios,' pesos');
    writeln('El salario del empleado más nuevo es ',maxSalario);
end.
```

Sin errores:

```
program TP2_E4;
{$codepage UTF8}
uses crt;
procedure leerDatos(var legajo: integer; var salario: real);
begin
    legajo:=1+random(high(integer));
    salario:=1+random(100);
end;
procedure actualizarMaximo(nuevoLegajo: integer; nuevoSalario: real; var maxLegajo: integer;
var maxSalario: real);
begin
    if (nuevoLegajo>maxLegajo) then
```

```
begin
  maxLegajo:=nuevoLegajo;
  maxSalario:=nuevoSalario;
end;
end;
var
  legajo, maxLegajo, i: integer;
  salario, maxSalario, sumaSalarios: real;
begin
  randomize;
  maxLegajo:=0; maxSalario:=0;
  sumaSalarios:=0;
  for i:= 1 to 130 do
    begin
      leerDatos(legajo,salario);
      actualizarMaximo(legajo,salario,maxLegajo,maxSalario);
      sumaSalarios:=sumaSalarios+salario;
    end;
    textcolor(green); write('En todo el mes, se gastan '); textcolor(red);
write(sumaSalarios:0:2); textcolor(green); writeln(' pesos');
    textcolor(green); write('El salario del empleado más nuevo es '); textcolor(red);
write(maxSalario:0:2);
end.
```

Los 5 errores que posee el código son:

1. Línea 2: En el *procedure* “*leerDatos*”, el parámetro “*salario*” debe ser por referencia.
2. Línea 8: En el *procedure* “*actualizarMaximo*”, falta pasar la variable local al programa “*maxSalario*” como parámetro por referencia, por lo que no se debe crear la variable local al proceso homónima.
3. Línea 19: En el programa principal, falta crear la variable local al programa “*sumaSalarios*”.
4. Línea 21: En el programa principal, falta inicializar las variables locales al programa “*maxLegajo*” y “*maxSalario*”.
5. Línea 23: En el programa principal, en la invocación del *procedure* “*leerDatos*”, el orden de los parámetros está invertido.



**Ejercicio 5.**

(a) Realizar un módulo que reciba un par de números (numA, numB) y retorne si numB es el doble de numA.

(b) Utilizando el módulo realizado en el inciso (a), realizar un programa que lea secuencias de pares de números hasta encontrar el par (0,0), e informe la cantidad total de pares de números leídos y la cantidad de pares en las que numB es el doble de numA. Ejemplo: si se lee la siguiente secuencia (1,2) (3,4) (9,3) (7,14) (0,0), el programa debe informar los valores 4 (cantidad de pares leídos) y 2 (cantidad de pares en los que numB es el doble de numA).

```
program TP2_E5;
{$codepage UTF8}
uses crt;
const
    numA_salida=0; numB_salida=0;
    multiplo_corte=2;
procedure leer_numeros(var numA, numB: int16);
begin
    numA:=numA_salida+random(101);
    numB:=numB_salida+random(101);
end;
function cumple_criterio(numA, numB: int16): boolean;
begin
    cumple_criterio:=(numB=multiplo_corte*numA);
end;
var
    numA, numB: int16;
    pares_total, pares_doble: int32;
begin
    randomize;
    pares_total:=0; pares_doble:=0;
    leer_numeros(numA,numB);
    while ((numA<>numA_salida) or (numB<>numB_salida)) do
    begin
        pares_total:=pares_total+1;
        if (cumple_criterio(numA,numB)=true) then
            pares_doble:=pares_doble+1;
            leer_numeros(numA,numB);
        end;
        textcolor(green); write('La cantidad total de pares leídos es '); textcolor(red);
        writeln(pares_total);
        textcolor(green); write('La cantidad de pares en las que numB es el doble de numA es ');
        textcolor(red); write(pares_doble);
    end.
```

## Ejercicio 6.

Realizar un programa modularizado que lea datos de 100 productos de una tienda de ropa. Para cada producto, debe leer el precio, código y tipo (pantalón, remera, camisa, medias, campera, etc.). Informar:

- Código de los dos productos más baratos.
- Código del producto de tipo “pantalón” más caro.
- Precio promedio.

```

program TP2_E6;
{$codepage UTF8}
uses crt;
const
    productos_total=100;
    tipo_corte='pantalon';
procedure leer_producto(var precio: real; var producto: int16; var tipo: string);
var
    vector_tipos: array[1..5] of string=('pantalon', 'remera', 'camisa', 'medias', 'campera');
begin
    precio:=1+random(100);
    producto:=1+random(100);
    tipo:=vector_tipos[1+random(5)];
end;
procedure actualizar_minimos(precio: real; producto: int16; var precio_min1, precio_min2:
real; var producto_min1, producto_min2: int16);
begin
    if (precio<precio_min1) then
    begin
        precio_min2:=precio_min1;
        producto_min2:=producto_min1;
        precio_min1:=precio;
        producto_min1:=producto;
    end
    else
        if (precio<precio_min2) then
        begin
            precio_min2:=precio;
            producto_min2:=producto;
        end;
    end;
end;
procedure actualizar_maximo(precio: real; producto: int16; tipo: string; var precio_max: real;
var producto_max: int16);
begin
    if ((tipo=tipo_corte) and (precio>precio_max)) then
    begin
        precio_max:=precio;
        producto_max:=producto;
    end;
end;
procedure leer_productos(var producto_min1, producto_min2, producto_max: int16; var
precio_prom: real);
var
    i: int8;
    producto: int16;
    precio, precio_min1, precio_min2, precio_max, precio_total: real;
    tipo: string;
begin
    precio_min1:=9999999; precio_min2:=9999999;
    precio_max:=-9999999;
    precio_total:=0;

```

```
for i:= 1 to productos_total do
begin
    leer_producto(precio,producto,tipo);
    actualizar_minimos(precio,producto,precio_min1,precio_min2,producto_min1,producto_min2);
    actualizar_maximo(precio,producto,tipo,precio_max,producto_max);
    precio_total:=precio_total+precio;
end;
precio_prom:=precio_total/productos_total;
end;
var
    producto_min1, producto_min2, producto_max: int16;
    precio_prom: real;
begin
    randomize;
    producto_min1:=0; producto_min2:=0;
    producto_max:=0;
    precio_prom:=0;
    leer_productos(producto_min1,producto_min2,producto_max,precio_prom);
    textcolor(green); write('Los códigos de los dos productos más baratos son ');
    textcolor(red); write(producto_min1); textcolor(green); write(' y '); textcolor(red);
    writeln(producto_min2);
    textcolor(green); write('El código del producto de tipo '); textcolor(yellow);
    write(tipo_corte); textcolor(green); write(' más caro es '); textcolor(red);
    writeln(producto_max);
    textcolor(green); write('El precio promedio es $'); textcolor(red); write(precio_prom:0:2);
end.
```

## Ejercicio 7.

(a) Realizar un módulo que reciba como parámetro un número entero y retorne la cantidad de dígitos que posee y la suma de los mismos.

(b) Utilizando el módulo anterior, realizar un programa que lea una secuencia de números e imprima la cantidad total de dígitos leídos. La lectura finaliza al leer un número cuyos dígitos suman, exactamente, 10, el cual debe procesarse.

```
program TP2_E7;
{$codepage UTF8}
uses crt;
const
    suma_salida=10;
procedure cantidad_suma_digitos(num: int16; var digitos, suma: int16);
var
    digito: int8;
begin
    while (num>0) do
    begin
        digito:=num mod 10;
        digitos:=digitos+1;
        suma:=suma+digito;
        num:=num div 10;
    end;
end;
procedure cantidad_digitos_total(var digitos_total: int16);
var
    num, digitos, suma: int16;
begin
    repeat
        num:=1+random(1000);
        digitos:=0; suma:=0;
        cantidad_suma_digitos(num,digitos,suma);
        digitos_total:=digitos_total+digitos;
    until (suma=suma_salida);
end;
var
    digitos_total: int16;
begin
    randomize;
    digitos_total:=0;
    cantidad_digitos_total(digitos_total);
    textcolor(green); write('La cantidad total de dígitos leídos es '); textcolor(red);
    write(digitos_total);
end.
```

## Ejercicio 8.

Realizar un programa modularizado que lea secuencia de números enteros. La lectura finaliza cuando llega el número 123456, el cual no debe procesarse. Informar en pantalla, para cada número, la suma de sus dígitos pares y la cantidad de dígitos impares que posee.

```
program TP2_E8;
{$codepage UTF8}
uses crt;
const
    num_salida=123456;
procedure suma_pares_cantidad_impares(num: int32; var suma_pares, cantidad_impares: int16);
var
    digito: int8;
begin
    while (num>0) do
    begin
        digito:=num mod 10;
        if (digito mod 2=0) then
            suma_pares:=suma_pares+digito
        else
            cantidad_impares:=cantidad_impares+1;
        num:=num div 10;
    end;
end;
var
    suma_pares, cantidad_impares: int16;
    num: int32;
begin
    randomize;
    num:=num_salida+random(100);
    while (num<>num_salida) do
    begin
        suma_pares:=0; cantidad_impares:=0;
        suma_pares_cantidad_impares(num,suma_pares,cantidad_impares);
        textcolor(green); write('La suma de los dígitos pares del número '); textcolor(yellow);
        write(num); textcolor(green); write(' es '); textcolor(red); writeln(suma_pares);
        textcolor(green); write('La cantidad de dígitos impares que posee el número ');
        textcolor(yellow); write(num); textcolor(green); write(' es '); textcolor(red);
        writeln(cantidad_impares);
        num:=num_salida+random(100);
    end;
end.
```

## Ejercicio 9.

Realizar un programa modularizado que lea información de alumnos de una facultad. Para cada alumno, se lee: número de inscripción, apellido y nombre. La lectura finaliza cuando se ingresa el alumno con número de inscripción 1200, que debe procesarse. Se pide calcular e informar:

- Apellido de los dos alumnos con número de inscripción más chico.
- Nombre de los dos alumnos con número de inscripción más grande.
- Porcentaje de alumnos con número de inscripción par.

```

program TP2_E9;
{$codepage UTF8}
uses crt;
const
    numero_salida=1200;
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_alumno(var numero: int16; var apellido, nombre: string);
var
    i: int8;
begin
    i:=random(100);
    if (i=0) then
        numero:=numero_salida
    else
        numero:=1+random(random(high(int16)));
        apellido:=random_string(5+random(6));
        nombre:=random_string(5+random(6));
    end;
end;
procedure actualizar_minimos(numero: int16; apellido: string; var numero_min1, numero_min2:
int16; var apellido_min1, apellido_min2: string);
begin
    if (numero<numero_min1) then
        begin
            numero_min2:=numero_min1;
            apellido_min2:=apellido_min1;
            numero_min1:=numero;
            apellido_min1:=apellido;
        end
    else
        if (numero<numero_min2) then
            begin
                numero_min2:=numero;
                apellido_min2:=apellido;
            end;
    end;
end;
procedure actualizar_maximos(numero: int16; nombre: string; var numero_max1, numero_max2:
int16; var nombre_max1, nombre_max2: string);
begin
    if (numero>numero_max1) then
        begin
            numero_max2:=numero_max1;

```

```
nombre_max2:=nombre_max1;
numero_max1:=numero;
nombre_max1:=nombre;
end
else
  if (numero>numero_max2) then
    begin
      numero_max2:=numero;
      nombre_max2:=nombre;
    end;
end;
end;

procedure leer_alumnos(var apellido_min1, apellido_min2, nombre_max1, nombre_max2: string; var
porcentaje_par: real);
var
  alumnos_par, alumnos_total: int16;
  numero, numero_min1, numero_min2, numero_max1, numero_max2: int16;
  apellido, nombre: string;
begin
  alumnos_par:=0; alumnos_total:=0;
  numero_min1:=high(int16); numero_min2:=high(int16);
  numero_max1:=low(int16); numero_max2:=low(int16);
  repeat
    leer_alumno(numero,apellido,nombre);
    actualizar_minimos(numero,apellido,numero_min1,numero_min2,apellido_min1,apellido_min2);
    actualizar_maximos(numero,nombre,numero_max1,numero_max2,nombre_max1,nombre_max2);
    alumnos_total:=alumnos_total+1;
    if (numero mod 2=0) then
      alumnos_par:=alumnos_par+1;
    until (numero=numero_salida);
    porcentaje_par:=alumnos_par/alumnos_total*100;
end;
var
  porcentaje_par: real;
  apellido_min1, apellido_min2, nombre_max1, nombre_max2: string;
begin
  randomize;
  apellido_min1:=''; apellido_min2:='';
  nombre_max1:=''; nombre_max2:='';
  porcentaje_par:=0;
  leer_alumnos(apellido_min1,apellido_min2,nombre_max1,nombre_max2,porcentaje_par);
  textcolor(green); write('Los apellidos de los dos alumnos con número de inscripción más
chico son '); textcolor(red); write(apellido_min1); textcolor(green); write(' y ');
textcolor(red); writeln(apellido_min2);
  textcolor(green); write('Los nombres de los dos alumnos con número de inscripción más grande
son '); textcolor(red); write(nombre_max1); textcolor(green); write(' y '); textcolor(red);
writeln(nombre_max2);
  textcolor(green); write('El porcentaje de alumnos con número de inscripción par es ');
textcolor(red); write(porcentaje_par:0:2); textcolor(green); write('%');
end.
```

## Ejercicio 10.

Realizar un programa modularizado que lea una secuencia de caracteres y verifique si cumple con el patrón A\$B#, donde:

- A es una secuencia de sólo letras vocales.
- B es una secuencia de sólo caracteres alfabéticos sin letras vocales.
- Los caracteres \$ y # seguro existen.

*Nota: En caso de no cumplir, informar qué parte del patrón no se cumplió.*

```

program TP2_E10;
{$codepage UTF8}
uses crt;
const
  caracter_corte1='$'; caracter_corte2='#';
function es_vocal(c: char): boolean;
begin
  es_vocal:=(c='A') or (c='E') or (c='I') or (c='O') or (c='U') or (c='a') or (c='e') or
(c='i') or (c='o') or (c='u');
end;
function cumple_secuenciaA(cumple: boolean): boolean;
var
  c: char;
begin
  textcolor(green); writeln('Introducir caracter de la secuencia A:');
  textcolor(yellow); readln(c);
  while ((c<>caracter_corte1) and (cumple=true)) do
  begin
    if (es_vocal(c)=false) then
      cumple:=false
    else
      begin
        textcolor(yellow); readln(c);
      end;
    end;
  end;
  cumple_secuenciaA:=cumple;
end;
function cumple_secuenciaB(cumple: boolean): boolean;
var
  c: char;
begin
  textcolor(green); writeln('Introducir caracter de la secuencia B:');
  textcolor(yellow); readln(c);
  while ((c<>caracter_corte2) and (cumple=true)) do
  begin
    if (es_vocal(c)=true) then
      cumple:=false
    else
      begin
        textcolor(yellow); readln(c);
      end;
    end;
  end;
  cumple_secuenciaB:=cumple;
end;
var
  cumple: boolean;
begin
  cumple:=true;
  cumple:=cumple_secuenciaA(cumple);
  if (cumple=true) then

```



```
begin
  cumple:=cumple_secuenciaB(cumple);
  if (cumple=true) then
    begin
      textcolor(red); write('La secuencia cumple con el patrón A$B#');
    end
  else
    begin
      textcolor(red); write('La secuencia no cumple con la parte B del patrón A$B#');
    end;
  end
else
  begin
    textcolor(red); write('La secuencia no cumple con la parte A del patrón A$B#');
  end;
end.
```

## Ejercicio 11.

Realizar un programa modularizado que lea una secuencia de caracteres y verifique si cumple con el patrón  $A\%B^*$ , donde:

- $A$  es una secuencia de caracteres en la que no existe el carácter '\$'.
- $B$  es una secuencia con la misma cantidad de caracteres que aparecen en  $A$  y en la que aparece, a lo sumo, 3 veces el carácter '@'.
- Los caracteres % y \* seguro existen.

Nota: En caso de no cumplir, informar que parte del patrón no se cumplió.

```

program TP2_E11;
{$codepage UTF8}
uses crt;
const
    caracter_corte1='%'; caracter_corte2='*';
    caracter_corte3='$';
    caracter_corte4='@'; cantidad_caracter_corte4=3;
procedure cumple_secuenciaA(var cumple: boolean; var caracteresA: int8);
var
    c: char;
begin
    textcolor(green); writeln('Introducir caracter de la secuencia A:');
    textcolor(yellow); readln(c);
    while ((c<>caracter_corte1) and (cumple=true)) do
    begin
        if (c=caracter_corte3) then
            cumple:=false
        else
            begin
                caracteresA:=caracteresA+1;
                textcolor(yellow); readln(c);
            end;
        end;
    end;
end;
procedure cumple_secuenciaB(var cumple: boolean; caracteresA: int8);
var
    caracteresB, caracteres_corte4: int8;
    c: char;
begin
    caracteresB:=0; caracteres_corte4:=0;
    textcolor(green); writeln('Introducir caracter de la secuencia B:');
    textcolor(yellow); readln(c);
    while ((c<>caracter_corte2) and (cumple=true)) do
    begin
        caracteresB:=caracteresB+1;
        if (c=caracter_corte4) then
            caracteres_corte4:=caracteres_corte4+1;
        if ((caracteresB>caracteresA) or (caracteres_corte4>cantidad_caracter_corte4)) then
            cumple:=false
        else
            begin
                textcolor(yellow); readln(c);
            end;
        end;
    end;
end;
var
    cumple: boolean;
    caracteresA: int8;
begin

```

```
cumple:=true; caracteresA:=0;
cumple_secuenciaA(cumple,caracteresA);
if (cumple=true) then
begin
  cumple_secuenciaB(cumple,caracteresA);
  if (cumple=true) then
  begin
    textcolor(red); write('La secuencia cumple con el patrón A%B*');
  end
  else
  begin
    textcolor(red); write('La secuencia no cumple con la parte B del patrón A%B*');
  end;
end
else
begin
  textcolor(red); write('La secuencia no cumple con la parte A del patrón A%B*');
end;
end.
```

**Ejercicio 12.**

(a) Realizar un módulo que calcule el rendimiento económico de una plantación de soja. El módulo debe recibir la cantidad de hectáreas (ha) sembradas, el tipo de zona de siembra (1: zona muy fértil, 2: zona estándar, 3: zona árida) y el precio en U\$S de la tonelada de soja; y devolver el rendimiento económico esperado de dicha plantación. Para calcular el rendimiento económico esperado, debe considerar el siguiente rendimiento por tipo de zona:

Tipo de zona	Rendimiento por ha
1	6 toneladas por ha
2	2,6 toneladas por ha
3	1,4 toneladas por ha

(b) ARBA desea procesar información obtenida de imágenes satelitales de campos sembrados con soja en la provincia de Buenos Aires. De cada campo, se lee: localidad, cantidad de hectáreas sembradas y el tipo de zona (1, 2 o 3). La lectura finaliza al leer un campo de 900 ha en la localidad 'Saladillo', que debe procesarse. El precio de la soja es de U\$S 320 por tn. Informar:

- La cantidad de campos de la localidad Tres de Febrero con rendimiento estimado superior a U\$S 10.000.
- La localidad del campo con mayor rendimiento económico esperado.
- La localidad del campo con menor rendimiento económico esperado.
- El rendimiento económico promedio.

```

program TP2_E12;
{$codepage UTF8}
uses crt;
const
  zona_ini=1; zona_fin=3;
  ha_salida=900; localidad_salida='Saladillo';
  precio=320.0;
  localidad_corte='Tres de Febrero'; rendimiento_corte=10000.0;
type
  t_zona=zona_ini..zona_fin;
function rendimiento_economico(ha: int16; zona: t_zona): real;
var
  vector_precios: array[t_zona] of real=(6, 2.6, 1.4);
begin
  rendimiento_economico:=ha*vector_precios[zona]*precio;
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_campo(var localidad: string; var ha: int16; var zona: t_zona);
var
  i: int8;
begin

```

```
i:=random(101);
if (i=0) then
begin
    localidad:=localidad_salida;
    ha:=ha_salida;
end
else
begin
    if (i<=50) then
        localidad:=localidad_corte
    else
        localidad:=random_string(5+random(6));
        ha:=1+random(100);
    end;
    zona:=zona_ini+random(zona_fin);
end;
procedure actualizar_maximo(rendimiento: real; localidad: string; var rendimiento_max: real;
var localidad_max: string);
begin
    if (rendimiento>rendimiento_max) then
    begin
        rendimiento_max:=rendimiento;
        localidad_max:=localidad;
    end;
end;
procedure actualizar_minimo(rendimiento: real; localidad: string; var rendimiento_min: real;
var localidad_min: string);
begin
    if (rendimiento<rendimiento_min) then
    begin
        rendimiento_min:=rendimiento;
        localidad_min:=localidad;
    end;
end;
procedure leer_campos(var campos_corte: int16; var rendimiento_prom: real; var localidad_max,
localidad_min: string);
var
    zona: t_zona;
    ha, campos_total: int16;
    rendimiento, rendimiento_max, rendimiento_min, rendimiento_total: real;
    localidad: string;
begin
    rendimiento_max:=-9999999;
    rendimiento_min:=9999999;
    rendimiento_total:=0; campos_total:=0;
    repeat
        leer_campo(localidad,ha,zona);
        rendimiento:=rendimiento_economico(ha,zona);
        rendimiento_total:=rendimiento_total+rendimiento;
        campos_total:=campos_total+1;
        if ((localidad=localidad_corte) and (rendimiento>rendimiento_corte)) then
            campos_corte:=campos_corte+1;
        actualizar_maximo(rendimiento,localidad,rendimiento_max,localidad_max);
        actualizar_minimo(rendimiento,localidad,rendimiento_min,localidad_min);
    until ((localidad=localidad_salida) and (ha=ha_salida));
    rendimiento_prom:=rendimiento_total/campos_total;
end;
var
    campos_corte: int16;
    rendimiento_prom: real;
    localidad_max, localidad_min: string;
begin
    randomize;
    campos_corte:=0;
    localidad_max:=''; localidad_min:='';
    rendimiento_prom:=0;
```

```
leer_campos(campos_corte,rendimiento_prom,localidad_max,localidad_min);
    textcolor(green); write('La cantidad de campos de la localidad '); textcolor(yellow);
write(localidad_corte); textcolor(green); write(' con rendimiento estimado superior a U$S ');
textcolor(yellow); write(rendimiento_corte:0:2); textcolor(green); write(' es ');
textcolor(red); writeln(campos_corte);
    textcolor(green); write('La localidad del campo con mayor rendimiento económico esperado es
'); textcolor(red); writeln(localidad_max);
    textcolor(green); write('La localidad del campo con menor rendimiento económico esperado es
'); textcolor(red); writeln(localidad_min);
    textcolor(green); write('El rendimiento económico promedio es U$D '); textcolor(red);
write(rendimiento_prom:0:2);
end.
```

### Ejercicio 13.

*Dado el siguiente programa:*

```
program TP2_E13;
{$codepage UTF8}
uses crt;
procedure intercambio(var num1, num2: integer);
var
    aux: integer;
begin
    aux:=num1;
    num1:=num2;
    num2:=aux;
end;
procedure sumar(num1: integer; var num2: integer);
begin
    num2:=num1+num2;
end;
var
    i, num1, num2: integer;
begin
    read(num1);
    read(num2);
    for i:= 1 to 3 do
    begin
        intercambio(num1,num2);
        sumar(i,num1);
    end;
    writeln(num1);
end.
```

(a) ¿Qué imprime si se leen los valores  $num1 = 10$  y  $num2 = 5$ ?

Si se leen, los valores  $num1 = 10$  y  $num2 = 5$ , el programa imprime 9.

(b) ¿Qué imprime si se leen los valores  $num1 = 5$  y  $num2 = 10$ ?

Si se leen, los valores  $num1 = 5$  y  $num2 = 10$ , el programa imprime 14.

## Ejercicio 14.

Realizar un programa modularizado que lea 10 pares de números ( $X$ ,  $Y$ ) e informe, para cada par de números, la suma y el producto de todos los números entre  $X$  e  $Y$ . Por ejemplo, dado el par (3, 6), debe informar:

“La suma es 18” (obtenido de calcular  $3+4+5+6$ ).

“El producto es 360” (obtenido de calcular  $3*4*5*6$ ).

```
program TP2_E14;
{$codepage UTF8}
uses crt;
const
    pares_total=10;
procedure leer_numeros(var numX, numY: int8);
begin
    numX:=1+random(10);
    numY:=1+random(10);
end;
procedure verificar_numeros(var numX, numY: int8);
var
    aux: int8;
begin
    if (numX>numY) then
    begin
        aux:=numX;
        numX:=numY;
        numY:=aux;
    end;
end;
procedure calcular_suma_producto(var numX, numY: int8; var suma, producto: real);
var
    i: int8;
begin
    for i:= numX to numY do
    begin
        suma:=suma+i;
        producto:=producto*i;
    end;
end;
var
    i: int8;
    numX, numY: int8;
    suma, producto: real;
begin
    randomize;
    for i:= 1 to pares_total do
    begin
        leer_numeros(numX,numY);
        verificar_numeros(numX,numY);
        suma:=0; producto:=1;
        calcular_suma_producto(numX,numY,suma,producto);
        textcolor(green); write('Para el par '); textcolor(yellow); write('(' ,numX, ', ', numY, ')');
        textcolor(green); write(', la suma es '); textcolor(red); write(suma:0:2); textcolor(green);
        write(' y el producto es '); textcolor(red); writeln(producto:0:2);
    end;
end.
```



## Ejercicio 15.

Realizar un programa modularizado que lea información de 200 productos de un supermercado. De cada producto, se lee código y precio (cada código es un número entre 1 y 200). Informar en pantalla:

- Los códigos de los dos productos más baratos.
- La cantidad de productos de más de 16 pesos con código par.

```

program TP2_E15;
{$codepage UTF8}
uses crt;
const
  producto_ini=1; producto_fin=200;
  precio_corte=16.0;
type
  t_producto=producto_ini..producto_fin;
procedure leer_producto(var producto: t_producto; var precio: real);
begin
  producto:=producto_ini+random(producto_fin);
  precio:=1+random(100);
end;
procedure actualizar_minimos(precio: real; producto: t_producto; var precio_min1, precio_min2:
real; var producto_min1, producto_min2: int16);
begin
  if (precio<precio_min1) then
  begin
    precio_min2:=precio_min1;
    producto_min2:=producto_min1;
    precio_min1:=precio;
    producto_min1:=producto;
  end
  else
    if (precio<precio_min2) then
    begin
      precio_min2:=precio;
      producto_min2:=producto;
    end;
  end;
end;
procedure leer_productos(var producto_min1, producto_min2, productos_corte: int16);
var
  i, producto: t_producto;
  precio, precio_min1, precio_min2: real;
begin
  precio_min1:=9999999; precio_min2:=9999999;
  for i:= producto_ini to producto_fin do
  begin
    leer_producto(producto,precio);
    actualizar_minimos(precio,producto,precio_min1,precio_min2,producto_min1,producto_min2);
    if ((precio>precio_corte) and (producto mod 2=0)) then
      productos_corte:=productos_corte+1;
    end;
  end;
end;
var
  producto_min1, producto_min2, productos_corte: int16;
begin
  randomize;
  producto_min1:=0; producto_min2:=0;
  productos_corte:=0;
  leer_productos(producto_min1,producto_min2,productos_corte);

```

```
    textcolor(green); write('Los códigos de los dos productos más baratos son ');
textcolor(red); write(producto_min1); textcolor(green); write(' y '); textcolor(red);
writeln(producto_min2);
    textcolor(green); write('La cantidad de productos de más de '); textcolor(yellow);
write(precio_corte:0:2); textcolor(green); write(' pesos con código par es '); textcolor(red);
write(productos_corte);
end.
```

**Ejercicio 16.**

(a) Realizar un módulo que reciba como parámetro el radio de un círculo y retorne su diámetro y su perímetro.

(b) Utilizando el módulo anterior, realizar un programa que analice información de planetas obtenida del Telescopio Espacial Kepler. De cada planeta, se lee su nombre, su radio (medido en kilómetros) y la distancia (medida en años luz) a la Tierra. La lectura finaliza al leer un planeta con radio 0, que no debe procesarse. Informar:

- Nombre y distancia de los planetas que poseen un diámetro menor o igual que el de la Tierra (12.700 km) y mayor o igual que el de Marte (6.780 km).
- Cantidad de planetas con un perímetro superior al del planeta Júpiter (439.264 km).

```

program TP2_E16;
{$codepage UTF8}
uses crt;
const
  radio_salida=0.0;
  diametro_corte1=12700.0; diametro_corte2=6780.0;
  perimetro_corte=439264.0;
procedure circulo(radio: real; var diametro, perimetro: real);
begin
  diametro:=radio*2;
  perimetro:=pi*diametro;
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_planeta(var nombre: string; var radio, distancia: real);
var
  i: int8;
begin
  nombre:=random_string(5+random(6));
  i:=random(100);
  if (i=0) then
    radio:=radio_salida
  else
    radio:=1+random(100000);
    distancia:=1+random(100);
  end;
end;
procedure leer_planetas(var planetas_corte: int16);
var
  radio, distancia, diametro, perimetro: real;
  nombre: string;
begin
  diametro:=0; perimetro:=0;
  leer_planeta(nombre,radio,distancia);
  while (radio<>radio_salida) do
    begin
      circulo(radio,diametro,perimetro);
      if ((diametro<=diametro_corte1) and (diametro>=diametro_corte2)) then
        begin

```

```
    textcolor(green); write('El planeta '); textcolor(red); write(nombre); textcolor(green);
write(' tiene un diámetro menor o igual al de la Tierra ('); textcolor(yellow);
write(diametro_corte1:0:2); textcolor(green); write(' km) y mayor o igual que el de Marte (');
textcolor(yellow); write(diametro_corte2:0:2); textcolor(green); write(' km), y queda a ');
textcolor(red); write(distancia:0:2); textcolor(green); writeln(' años luz de la Tierra');
    end;
    if (perimetro>perimetro_corte) then
        planetas_corte:=planetas_corte+1;
        leer_planeta(nombre,radio,distancia);
    end;
end;
var
    planetas_corte: int16;
begin
    randomize;
    planetas_corte:=0;
    leer_planetas(planetas_corte);
    textcolor(green); write('La cantidad de planetas con un perímetro superior al del planeta
Júpiter ('); textcolor(yellow); write(perimetro_corte:0:2); textcolor(green); write(' km) es
'); textcolor(red); write(planetas_corte);
end.
```

## Ejercicio 17.

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.

(a) Analizar las soluciones a dichos problemas e identificar:

- (i) ¿Qué porciones de su código podrían modularizarse? ¿En qué casos propondría una estructura de módulos anidada?
- (ii) ¿Qué tipo de módulo (función o procedimiento) conviene utilizar en cada caso? ¿Existe algún caso en los que sólo un tipo de módulo es posible?
- (iii) ¿Qué mecanismos de comunicación conviene utilizar entre los módulos propuestos?

(b) Implementar, nuevamente, los 3 programas, teniendo en cuenta los módulos propuestos en el inciso anterior.

### Ejercicio 1:

```

program TP2_E17a;
{$codepage UTF8}
uses crt;
const
    empresa_salida=100;
    monto_corte=50000.0;
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 empresa, inversiones: int16; var monto_total: real);
var
    i: int8;
begin
    i:=random(100);
    if (i=0) then
        empresa:=empresa_salida
    else
        empresa:=1+random(high(int16));
        inversiones:=1+random(1000);
        leer_inversiones(empresa,inversiones,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
  empresa, inversiones: int16;
  monto_total, monto_max: real;
begin
  monto_max := -9999999;
  repeat
    leer_empresa(empresa, inversiones, monto_total);
    if (inversiones > 0) then
      begin
        calcular_a(empresa, inversiones, monto_total);
        calcular_b(monto_total, empresa, monto_max, empresa_max);
        calcular_c(monto_total, empresas_corte);
      end;
  until (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 TP2_E17b;
{$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;
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 legajo: int16; var condicion: char; var presente, nota_total,
notas_cero, notas_diez: int8);
var
    vector_condiciones: array[1..2] of char=(condicion_i, condicion_r);
    i: int8;
begin
    i:=random(100);
    if (i=0) then
        legajo:=legajo_salida
    else
        legajo:=1+random(high(int16));
    if (legajo<>legajo_salida) then
        begin
            condicion:=vector_condiciones[1+random(2)];
            leer_notas(presente,nota_total,notas_cero,notas_diez);
        end;
    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;
    end;
end;
procedure calcular_c(presente: int8; var alumnos_autoeva: int16);
begin
    if (presente=autoeva_total) then
        alumnos_autoeva:=alumnos_autoeva+1;
    end;
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;
end;
procedure calcular_e(notas_cero: int8; var alumnos_cero: int16);
begin
    if (notas_cero>=1) then
        alumnos_cero:=alumnos_cero+1;
    end;
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;
        end;
    end;
end;
```

```

    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;
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
    presente, nota_total, notas_cero, notas_diez, notas_diez_max1, notas_diez_max2,
notas_cero_max1, notas_cero_max2: int8;
    legajo: int16;
    condicion: char;
begin
    notas_diez_max1:=0; notas_diez_max2:=0;
    notas_cero_max1:=0; notas_cero_max2:=0;
    leer_alumno(legajo,condicion,presente,nota_total,notas_cero,notas_diez);
    while (legajo<>legajo_salida) do
    begin
        calcular_ab(condicion,presente,ingresantes_total,ingresantes_parcial,recursantes_total,rec
ursantes_parcial);
        calcular_c(presente,alumnos_autoeva);
        calcular_d(nota_total,alumnos_corte);
        calcular_e(notas_cero,alumnos_cero);
        calcular_f(notas_diez,legajo,notas_diez_max1,notas_diez_max2,legajo_diez_max1,legajo_diez_
max2);
        calcular_g(notas_cero,legajo,notas_cero_max1,notas_cero_max2,legajo_cero_max1,legajo_cero_
max2);
        leer_alumno(legajo,condicion,presente,nota_total,notas_cero,notas_diez);
    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 TP2_E17c;
{$codepage UTF8}
uses crt;
const
  tanque_r='R'; tanque_c='C';
  tanque_salida='Z';
  alto_corte=1.40;
  volumen_corte=800.0;
procedure leer_tanque(var tanque: char; var alto, volumen: real);
var
  vector_tanques: array[1..2] of char=(tanque_r, tanque_c);
  i: int8;
  radio, ancho, largo: real;
begin

```

```
i:=random(100);
if (i=0) then
    tanque:=tanque_salida
else
    tanque:=vector_tanques[1+random(2)];
if (tanque<>tanque_salida) then
begin
    if (tanque=tanque_r) then
    begin
        ancho:=1+random(391)/10;
        largo:=1+random(391)/10;
        alto:=1+random(21)/10;
        volumen:=ancho*largo*alto;
    end
    else
    begin
        radio:=1+random(391)/10;
        alto:=1+random(21)/10;
        volumen:=pi*radio*radio*alto;
    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
    volumen, alto: real;
    tanque: char;
begin
    leer_tanque(tanque,alto,volumen);
    while (tanque<>tanque_salida) do
    begin
```

```

    calcular_a(volumen,volumen_max1,volumen_max2);
    calcular_bc(tanque,volumen,volumen_total_c,volumen_total_r,tanques_c,tanques_r);
    calcular_d(alto,tanques_corte_alto);
    calcular_e(volumen,tanques_corte_volumen);
    leer_tanque(tanque,alto,volumen);
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.

```