**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.

**(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.

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.

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.

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;

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;

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;

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;

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;

function leer\_secuencia(secuencia: string): string;

begin

  textcolor(green); write('Introducir secuencia de caracteres: ');

  textcolor(yellow); readln(secuencia);

  leer\_secuencia:=secuencia;

end;

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;

procedure parseo\_string(var cumple\_A, cumple\_B, cumple\_AB: boolean);

var

  i, j: int8;

  secuencia: string;

begin

  secuencia:='';

  secuencia:=leer\_secuencia(secuencia);

  i:=1;

  while (secuencia[i]<>'$') do

  begin

    cumple\_A:=cumple\_A and (es\_vocal(secuencia[i])=true);

    cumple\_AB:=cumple\_AB and (es\_vocal(secuencia[i])=true);

    i:=i+1;

  end;

  j:=i+1;

  while (secuencia[j]<>'#') do

  begin

    cumple\_B:=cumple\_B and (es\_vocal(secuencia[j])=false);

    cumple\_AB:=cumple\_AB and (es\_vocal(secuencia[j])=false);

    j:=j+1;

  end;

end;

var

  cumple\_A, cumple\_B, cumple\_AB: boolean;

begin

  cumple\_A:=true; cumple\_B:=true; cumple\_AB:=true;

  parseo\_string(cumple\_A,cumple\_B,cumple\_AB);

  if (cumple\_AB=true) then

  begin

    textcolor(yellow); write('La secuencia cumple con el patrón A$B#');

  end

  else if ((cumple\_A=false) and (cumple\_B=true)) then

  begin

    textcolor(yellow); write('La secuencia no cumple con la parte A del patrón A$B#');

  end

  else if ((cumple\_A=true) and (cumple\_B=false)) then

  begin

    textcolor(yellow); write('La secuencia no cumple con la parte B del patrón A$B#');

  end

  else

  begin

    textcolor(yellow); write('La secuencia no cumple con las partes A y B 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

  arrobas\_corte=3;

function leer\_secuencia(secuencia: string): string;

begin

  textcolor(green); write('Introducir secuencia de caracteres: ');

  textcolor(yellow); readln(secuencia);

  leer\_secuencia:=secuencia;

end;

procedure parseo\_string(var cumple\_A, cumple\_B, cumple\_AB: boolean);

var

  i, j, arrobas: int8;

  secuencia: string;

begin

  secuencia:='';

  secuencia:=leer\_secuencia(secuencia);

  i:=1; arrobas:=0;

  while (secuencia[i]<>'%') do

  begin

    cumple\_A:=cumple\_A and (secuencia[i]<>'$');

    cumple\_AB:=cumple\_AB and (secuencia[i]<>'$');

    i:=i+1;

  end;

  j:=i+1;

  while (secuencia[j]<>'\*') do

  begin

    if (secuencia[j]='@') then

      arrobas:=arrobas+1;

    cumple\_B:=cumple\_B and (arrobas<=arrobas\_corte);

    cumple\_AB:=cumple\_AB and (arrobas<=arrobas\_corte);

    j:=j+1;

  end;

  cumple\_B:=cumple\_B and (j/2=i);

  cumple\_AB:=cumple\_AB and (j/2=i);

end;

var

  cumple\_A, cumple\_B, cumple\_AB: boolean;

begin

  cumple\_A:=true; cumple\_B:=true; cumple\_AB:=true;

  parseo\_string(cumple\_A,cumple\_B,cumple\_AB);

  if (cumple\_AB=true) then

  begin

    textcolor(yellow); write('La secuencia cumple con el patrón A%B\*');

  end

  else if ((cumple\_A=false) and (cumple\_B=true)) then

  begin

    textcolor(yellow); write('La secuencia no cumple con la parte A del patrón A%B\*');

  end

  else if ((cumple\_A=true) and (cumple\_B=false)) then

  begin

    textcolor(yellow); write('La secuencia no cumple con la parte B del patrón A%B\*');

  end

  else

  begin

    textcolor(yellow); write('La secuencia no cumple con las partes A y B 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;

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;

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;

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;

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

begin

  if (condicion=condicion\_i) then

  begin

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

      ingresantes\_parcial:=ingresantes\_parcial+1;

    ingresantes\_total:=ingresantes\_total+1;

  end

  else

  begin

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

      recursantes\_parcial:=recursantes\_parcial+1;

    recursantes\_total:=recursantes\_total+1;

  end;

end;

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

begin

  if (presente=autoeva\_total) then

    alumnos\_autoeva:=alumnos\_autoeva+1;

end;

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

begin

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

    alumnos\_corte:=alumnos\_corte+1;

end;

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

begin

  if (notas\_cero>=1) then

    alumnos\_cero:=alumnos\_cero+1;

end;

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

begin

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

  begin

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

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

    notas\_diez\_max1:=notas\_diez;

    legajo\_diez\_max1:=legajo;

  end

  else

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

    begin

      notas\_diez\_max2:=notas\_diez;

      legajo\_diez\_max2:=legajo;

    end;

end;

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

begin

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

  begin

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

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

    notas\_cero\_max1:=notas\_cero;

    legajo\_cero\_max1:=legajo;

  end

  else

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

    begin

      notas\_cero\_max2:=notas\_cero;

      legajo\_cero\_max2:=legajo;

    end;

end;

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

var

  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,recursantes\_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,tanques\_corte\_alto,tanques\_corte\_volumen);

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

  begin

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

    if (tanques\_c>0) then

    begin

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

    end

    else

    begin

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

    end;

    if (tanques\_r>0) then

    begin

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

    end

    else

    begin

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

    end;

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

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

  end

  else

  begin

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

  end;

end.