**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 un 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?*

Si se lee la secuencia de valores 250, 35 y 100, se 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.*

program TP2\_E3;

*{$codepage UTF8}*

uses crt;

*{Suma los números entre a y b y retorna el resultado en c}*

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

  result:=0;

  readln(a); readln(b);

  sumar(a,b,result);

  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.

Los 6 errores que existen en el siguiente programa son:

1. Línea 3: Falta el “;” al final de la sentencia.
2. Línea 3: En el procedure “sumar”, el parámetro “c” debe ser de referencia.
3. Línea 8: En el procedure “sumar”, se debe crear la variable correspondiente al índice “i”.
4. Línea 9: En el procedure “sumar”, se debe inicializar la variable “suma”.
5. Línea 13: Falta crear las variables globales “a” y “b”, como integers, y “ok”, como boolean.
6. Línea 17: En el procedure “sumar”, se debe pasar la variable global “result” como parámetro de valor.

**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 mayor legajo.*

program TP2\_E4;

*{$codepage UTF8}*

uses crt;

procedure leerDatos(var legajo: integer; salario: real);

begin

  writeln('Ingresar 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.

Los 5 errores que posee el código son:

1. Línea 2: En el procedure “leerDatos”, el parámetro “salario” debe ser de referencia.
2. Línea 8: En el procedure “actualizarMaximo”, se debe pasar la variable global “maxSalario” como parámetro de referencia, por lo que no se debe crear la variable local homónima.
3. Línea 19: Falta crear la variable global “sumaSalarios”.
4. Línea 21: Falta inicializar las variables “maxLegajo” y “maxSalario”.
5. Línea 23: 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.*

function multiplo(numA, numB: int16): boolean;

begin

  multiplo:=(numB=multiplo\_2\*numA);

end;

**(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\_2=2;

function multiplo(numA, numB: int16): boolean;

begin

  multiplo:=(numB=multiplo\_2\*numA);

end;

var

  pares\_leidos, doble: int8;

  numA, numB: int16;

begin

  pares\_leidos:=0; doble:=0;

  textcolor(green); write('Introducir número entero A: ');

  textcolor(yellow); readln(numA);

  textcolor(green); write('Introducir número entero B: ');

  textcolor(yellow); readln(numB);

  while ((numA<>numA\_salida) or (numB<>numB\_salida)) do

  begin

    if (multiplo(numA,numB)=true) then

      doble:=doble+1;

    pares\_leidos:=pares\_leidos+1;

    textcolor(green); write('Introducir número entero A: ');

    textcolor(yellow); readln(numA);

    textcolor(green); write('Introducir número entero B: ');

    textcolor(yellow); readln(numB);

  end;

  textcolor(green); write('La cantidad total de pares leídos es '); textcolor(red); writeln(pares\_leidos);

  textcolor(green); write('La cantidad de pares en las que numB es el doble de numA es '); textcolor(red); write(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;

procedure leer\_producto(var precio, codigo: int16; var tipo: string);

begin

  textcolor(green); write('Introducir precio de producto: ');

  textcolor(yellow); readln(precio);

  textcolor(green); write('Introducir código de producto: ');

  textcolor(yellow); readln(codigo);

  textcolor(green); write('Introducir tipo de producto: ');

  textcolor(yellow); readln(tipo);

end;

procedure actualizar\_minimos(precio, codigo: int16; var precio\_min1, precio\_min2, codigo\_min1, codigo\_min2: int16);

begin

  if (precio<precio\_min1) then

  begin

    precio\_min2:=precio\_min1;

    codigo\_min2:=codigo\_min1;

    precio\_min1:=precio;

    codigo\_min1:=codigo;

  end

  else

    if (precio<precio\_min2) then

    begin

      precio\_min2:=precio;

      codigo\_min2:=codigo;

    end;

end;

procedure actualizar\_pantalon\_mas\_caro(precio, codigo: int16; tipo: string; var precio\_max, codigo\_max: int16);

begin

  if ((tipo='pantalon') and (precio>precio\_max)) then

    begin

      precio\_max:=precio;

      codigo\_max:=codigo;

    end;

end;

procedure leer\_productos(var codigo\_min1, codigo\_min2, codigo\_max: int16; var precio\_prom: real);

var

  i: int8;

  precio, codigo, precio\_min1, precio\_min2, precio\_max, precio\_sum: int16;

  tipo: string;

begin

  precio\_min1:=high(int16); precio\_min2:=high(int16); precio\_max:=low(int16); precio\_sum:=0;

  for i:= 1 to productos\_total do

  begin

    leer\_producto(precio,codigo,tipo);

    actualizar\_minimos(precio,codigo,precio\_min1,precio\_min2,codigo\_min1,codigo\_min2);

    actualizar\_pantalon\_mas\_caro(precio,codigo,tipo,precio\_max,codigo\_max);

    precio\_sum:=precio\_sum+precio;

  end;

  precio\_prom:=precio\_sum/100;

end;

var

  codigo\_min1, codigo\_min2, codigo\_max: int16;

  precio\_prom: real;

begin

  codigo\_min1:=0; codigo\_min2:=0; codigo\_max:=0; precio\_prom:=0;

  leer\_productos(codigo\_min1,codigo\_min2,codigo\_max,precio\_prom);

  textcolor(green); write('Los códigos de los dos productos más baratos son '); textcolor(red); write(codigo\_min1); textcolor(green); write(' y '); textcolor(red); writeln(codigo\_min2);

  textcolor(green); write('El código del producto de tipo "pantalón" más caro es '); textcolor(red); writeln(codigo\_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.*

procedure cantidad\_suma\_digitos(num: int16; var digitos, suma: int16);

var

  digito: int8;

begin

  digitos:=0; suma:=0;

  while (num>0) do

  begin

    digito:=num mod 10;

    digitos:=digitos+1;

    suma:=suma+digito;

    num:=num div 10;

  end;

end;

**(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

  digitos:=0; suma:=0;

  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

  digitos\_total:=0; suma:=0;

  repeat

    textcolor(green); write('Introducir número entero: ');

    textcolor(yellow); readln(num);

    cantidad\_suma\_digitos(num,digitos,suma);

    digitos\_total:=digitos\_total+digitos;

  until (suma=suma\_salida);

end;

var

  digitos\_total: int16;

begin

  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

  suma\_pares:=0; cantidad\_impares:=0;

  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

  suma\_pares:=0; cantidad\_impares:=0;

  textcolor(green); write('Introducir unúmero entero: ');

  textcolor(yellow); readln(num);

  while (num<>num\_salida) do

  begin

    suma\_pares\_cantidad\_impares(num,suma\_pares,cantidad\_impares);

    textcolor(green); write('La suma de sus dígitos pares es '); textcolor(red); writeln(suma\_pares);

    textcolor(green); write('La cantidad de dígitos impares que posee es '); textcolor(red); writeln(cantidad\_impares);

    textcolor(green); write('Introducir otro número entero: ');

    textcolor(yellow); readln(num);

  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

  num\_salida=1200;

procedure leer\_alumno(var num: int32; var apellido, nombre: string);

begin

  textcolor(green); write('Introducir número de inscripción: ');

  textcolor(yellow); readln(num);

  textcolor(green); write('Introducir apellido: ');

  textcolor(yellow); readln(apellido);

  textcolor(green); write('Introducir nombre: ');

  textcolor(yellow); readln(nombre);

end;

procedure actualizar\_minimos(num: int32; apellido: string; var num\_min1, num\_min2: int32; var apellido\_min1, apellido\_min2: string);

begin

  if (num<num\_min1) then

  begin

    num\_min2:=num\_min1;

    apellido\_min2:=apellido\_min1;

    num\_min1:=num;

    apellido\_min1:=apellido;

  end

  else

    if (num<num\_min2) then

    begin

      num\_min2:=num;

      apellido\_min2:=apellido;

    end;

end;

procedure actualizar\_maximos(num: int32; nombre: string; var num\_max1, num\_max2: int32; var nombre\_max1, nombre\_max2: string);

begin

  if (num>num\_max1) then

  begin

    num\_max2:=num\_max1;

    nombre\_max2:=nombre\_max1;

    num\_max1:=num;

    nombre\_max1:=nombre;

  end

  else

    if (num>num\_max2) then

    begin

      num\_max2:=num;

      nombre\_max2:=nombre;

    end;

end;

procedure leer\_alumnos(var apellido\_min1, apellido\_min2, nombre\_max1, nombre\_max2: string; var porcentaje: real);

var

  alumnos\_par, alumnos\_total: int16;

  num, num\_min1, num\_min2, num\_max1, num\_max2: int32;

  apellido, nombre: string;

begin

  num\_min1:=high(int16); num\_min2:=high(int16); num\_max1:=low(int16); num\_max2:=low(int16); alumnos\_par:=0; alumnos\_total:=0;

  repeat

    leer\_alumno(num,apellido,nombre);

    actualizar\_minimos(num,apellido,num\_min1,num\_min2,apellido\_min1,apellido\_min2);

    actualizar\_maximos(num,nombre,num\_max1,num\_max2,nombre\_max1,nombre\_max2);

    alumnos\_total:=alumnos\_total+1;

    if (num mod 2=0) then

      alumnos\_par:=alumnos\_par+1;

  until (num=num\_salida);

  porcentaje:=alumnos\_par/alumnos\_total\*100;

end;

var

  porcentaje: real;

  apellido\_min1, apellido\_min2, nombre\_max1, nombre\_max2: string;

begin

  porcentaje:=0;

  leer\_alumnos(apellido\_min1,apellido\_min2,nombre\_max1,nombre\_max2,porcentaje);

  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: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 que 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]);

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

    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;

  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;

  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;

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

  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;

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<=3);

    cumple\_AB:=cumple\_AB and (arrobas<=3);

    j:=j+1;

  end;

  cumple\_B:=cumple\_B and (arrobas<=3) and (j/2=i);

  cumple\_AB:=cumple\_AB and (arrobas<=3) 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;

  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;

  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;

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

  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 |

function rendimiento\_economico(ha, zona, precio: int16): real;

begin

  case zona of

    1: rendimiento\_economico:=ha\*6\*precio;

    2: rendimiento\_economico:=ha\*2.6\*precio;

    3: rendimiento\_economico:=ha\*1.4\*precio;

  end;

end;

**(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

  ha\_salida=900; localidad\_salida='Saladillo'; precio=320; rendimiento\_corte=10000.0;

function rendimiento\_economico(ha, zona, precio: int16): real;

begin

  case zona of

    1: rendimiento\_economico:=ha\*6\*precio;

    2: rendimiento\_economico:=ha\*2.6\*precio;

    3: rendimiento\_economico:=ha\*1.4\*precio;

  end;

end;

procedure leer\_campo(var localidad: string; var ha, zona: int16);

begin

  textcolor(green); write('Introducir localidad del campo: ');

  textcolor(yellow); readln(localidad);

  textcolor(green); write('Introducir cantidad de hectáreas sembradas del campo: ');

  textcolor(yellow); readln(ha);

  textcolor(green); write('Introducir tipo de zona del campo (1: zona muy fértil, 2: zona estándar, 3: zona árida): ');

  textcolor(yellow); readln(zona);

end;

procedure leer\_campos(var campos\_10000: int16; var rendimiento\_prom: real; var localidad\_max, localidad\_min: string);

var

  ha, zona, campos\_total: int16;

  rendimiento, rendimiento\_max, rendimiento\_min, rendimiento\_total: real;

  localidad: string;

begin

  campos\_total:=0; rendimiento:=0; rendimiento\_max:=0; rendimiento\_min:=999999; rendimiento\_total:=0;

  repeat

    leer\_campo(localidad,ha,zona);

    rendimiento:=rendimiento\_economico(ha,zona,precio);

    rendimiento\_total:=rendimiento\_total+rendimiento;

    campos\_total:=campos\_total+1;

    if ((localidad='Tres de Febrero') and (rendimiento>rendimiento\_corte)) then

      campos\_10000:=campos\_10000+1;

    if (rendimiento>rendimiento\_max) then

    begin

      rendimiento\_max:=rendimiento;

      localidad\_max:=localidad;

    end;

    if (rendimiento<rendimiento\_min) then

    begin

      rendimiento\_min:=rendimiento;

      localidad\_min:=localidad;

    end;

  until ((localidad=localidad\_salida) and (ha=ha\_salida));

  rendimiento\_prom:=rendimiento\_total/campos\_total;

end;

var

  campos\_10000: int16;

  rendimiento\_prom: real;

  localidad\_max, localidad\_min: string;

begin

  campos\_10000:=0; rendimiento\_prom:=0;

  leer\_campos(campos\_10000,rendimiento\_prom,localidad\_max,localidad\_min);

  textcolor(green); write('La cantidad de campos de la localidad Tres de Febrero con rendimiento estimado superior a U$S 10.000 es '); textcolor(red); writeln(campos\_10000);

  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 $'); 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, 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, 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: int16);

begin

  textcolor(green); write('Introducir número entero X: ');

  textcolor(yellow); readln(numX);

  textcolor(green); write('Introducir número entero Y: ');

  textcolor(yellow); readln(numY);

end;

procedure calcular\_suma\_producto(var numX, numY: int16; var suma, producto: real);

var

  i: int16;

begin

  suma:=0; producto:=1;

  leer\_numeros(numX,numY);

  for i:= numX to numY do

  begin

    suma:=suma+i;

    producto:=producto\*i;

  end;

end;

var

  i: int8;

  numX, numY: int16;

  suma, producto: real;

begin

  suma:=0; producto:=1;

  for i:= 1 to pares\_total do

  begin

    calcular\_suma\_producto(numX,numY,suma,producto);

    textcolor(green); write('Para el par '); textcolor(red); 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

  productos\_total=200; precio\_corte=16;

procedure leer\_producto(var codigo, precio: int16);

begin

  textcolor(green); write('Introducir código de producto: ');

  textcolor(yellow); readln(codigo);

  textcolor(green); write('Introducir precio de producto: ');

  textcolor(yellow); readln(precio);

end;

procedure actualizar\_minimos(precio, codigo: int16; var precio\_min1, precio\_min2, codigo\_min1, codigo\_min2: int16);

begin

  if (precio<precio\_min1) then

  begin

    precio\_min2:=precio\_min1;

    codigo\_min2:=codigo\_min1;

    precio\_min1:=precio;

    codigo\_min1:=codigo;

  end

  else

    if (precio<precio\_min2) then

    begin

      precio\_min2:=precio;

      codigo\_min2:=codigo;

    end;

end;

procedure leer\_productos(var codigo\_min1, codigo\_min2, productos\_16: int16);

var

  i, codigo, precio, precio\_min1, precio\_min2: int16;

begin

  for i:= 1 to productos\_total do

  begin

    leer\_producto(codigo,precio);

    actualizar\_minimos(precio,codigo,precio\_min1,precio\_min2,codigo\_min1,codigo\_min2);

    if ((precio>precio\_corte) and (codigo mod 2=0)) then

      productos\_16:=productos\_16+1;

  end

end;

var

  codigo\_min1, codigo\_min2, productos\_16: int16;

begin

  codigo\_min1:=0; codigo\_min2:=0; productos\_16:=0;

  leer\_productos(codigo\_min1,codigo\_min2,productos\_16);

  textcolor(green); write('Los códigos de los dos productos más baratos son '); textcolor(red); write(codigo\_min1); textcolor(green); write(' y '); textcolor(red); writeln(codigo\_min2);

  textcolor(green); write('La cantidad de productos de más de 16 pesos con código par es '); textcolor(red); write(productos\_16);

end.

**Ejercicio 16.**

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

procedure circulo(radio: real; var diametro, perimetro: real);

begin

  diametro:=radio\*2;

  perimetro:=pi\*diametro;

end;

**(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; diametro\_tierra=12700; perimetro\_jupiter=439264;

procedure circulo(radio: real; var diametro, perimetro: real);

begin

  diametro:=radio\*2;

  perimetro:=pi\*diametro;

end;

procedure leer\_planeta(var nombre: string; var radio, distancia: real);

begin

  textcolor(green); write('Introducir nombre del planeta: ');

  textcolor(yellow); readln(nombre);

  textcolor(green); write('Introducir radio del planeta (medido en kilómetros): ');

  textcolor(yellow); readln(radio);

  textcolor(green); write('Introducir distancia a la tierra (medida en años luz): ');

  textcolor(yellow); readln(distancia);

end;

procedure leer\_planetas(var planetas: int16);

var

  radio, distancia, diametro, perimetro: real;

  nombre: string;

begin

  radio:=0; distancia:=0; diametro:=0; perimetro:=0;

  leer\_planeta(nombre,radio,distancia);

  while (radio<>radio\_salida) do

  begin

    circulo(radio,diametro,perimetro);

    if (diametro<=diametro\_tierra) then

      textcolor(green); write('El planeta '); textcolor(red); write(nombre); textcolor(green); write(' tiene un diámetro menor o igual al de la Tierra (12.700 km), de la que queda a '); textcolor(red); write(distancia:0:2); textcolor(green); writeln(' años luz');

    if (perimetro>perimetro) then

      planetas:=planetas+1;

    leer\_planeta(nombre,radio,distancia);

  end;

end;

var

  planetas: int16;

begin

  planetas:=0;

  leer\_planetas(planetas);

  textcolor(green); write('La cantidad de planetas con un perímetro superior al del planeta Júpiter (439.264 km) es '); textcolor(red); write(planetas);

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 sus 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.*

program TP2\_E17b1;

*{$codepage UTF8}*

uses crt;

const

  codigo\_salida=100; monto\_corte=50000;

procedure empresa\_mayor\_inversion(monto\_sum: int32; codigo: int16; var monto\_max: int32; var codigo\_max: int16);

begin

  if (monto\_sum>monto\_max) then

  begin

    monto\_max:=monto\_sum;

    codigo\_max:=codigo;

  end;

end;

function empresas\_50000(monto\_sum: int32): int16;

begin

  if (monto\_sum>monto\_corte) then

    empresas\_50000:=1

  else

    empresas\_50000:=0;

end;

function promedio\_inversiones(monto\_sum: int32; inversiones: int16): real;

begin

  promedio\_inversiones:=0;

  promedio\_inversiones:=monto\_sum/inversiones;

end;

procedure leer\_empresas(var codigo\_max, codigos\_50000: int16);

var

  i, codigo, monto, inversiones: int8;

  monto\_sum, monto\_max: int32;

  monto\_prom: real;

begin

  monto\_max:=0;

  repeat

    textcolor(green); write('Introducir código de empresa: ');

    textcolor(yellow); readln(codigo);

    textcolor(green); write('Introducir cantidad de inversiones: ');

    textcolor(yellow); readln(inversiones);

    monto\_sum:=0; monto\_prom:=0;

    for i:= 1 to inversiones do

    begin

      textcolor(green); write('Introducir monto de la inversión ', i, ': ');

      textcolor(yellow); readln(monto);

      monto\_sum:=monto\_sum+monto;

    end;

    empresa\_mayor\_inversion(monto\_sum,codigo,monto\_max,codigo\_max);

    codigos\_50000:=codigos\_50000+empresas\_50000(monto\_sum);

    monto\_prom:=promedio\_inversiones(monto\_sum,inversiones);

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

  until (codigo=codigo\_salida);

end;

var

  codigo\_max, codigos\_50000: int16;

begin

  codigo\_max:=0; codigos\_50000:=0;

  leer\_empresas(codigo\_max,codigos\_50000);

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

  textcolor(green); write('La cantidad de empresas con inversiones de más de $50.000 es '); textcolor(red); write(codigos\_50000);

end.