CADP2024

**Práctica2(parte2)** – Funciones, procedimientos y parámetros

**1.** Responda las preguntas en relación al siguiente programa:

**program** Ejercicio3;

**procedure** suma(num1: integer; **var** num2:integer); **begin**

num2 := num1+num2;

num1 := 0;

**end**;

**var** i, x:integer;

**begin**

read(x); *{ leo la variable x }* **for** i:= 1 to 5 **do** suma(i,x);

write(x); *{ imprimo las variable x }*

**end**.

1. ¿Qué imprime si se lee el valor 10 en la variable *x* ?

El programa imprime 25

1. 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);

**el programa imprime 10**

1. Qué sucede si se cambia el encabezado del *procedure* por:

**procedure** suma(**var** num1: integer; **var** num2:integer);

**se le quiere asignar un valor en la dirección de memoria de i**

**2.** Responda la pregunta en relación al siguiente programa:

**program** ejercicio4; **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, ‘Ímpares:’, impar);

**end**.

1. ¿Qué imprime si se lee la siguiente secuencia de valores? 250, 35, 100

4 pares 4 impares

1. 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:
   1. **program** ejercicio5;
   2. *{ suma los números entre a y b, y retorna el resultado en c }*
   3. **procedure** sumar(a, b, c : integer)
   4. **var**
   5. suma : integer;
   6. **begin**

7.

* 1. **for** i := a to b **do**
  2. suma := suma + i;
  3. c := c + suma;
  4. **end**;
  5. **var**
  6. result : integer;
  7. **begin**
  8. result := 0;
  9. readln(a); readln(b);
  10. sumar(a, b, 0);
  11. write(‘La suma total es ‘,result);
  12. *{ averigua si el resultado final estuvo entre 10 y 30}*
  13. ok := (result >= 10) **or** (result <= 30);
  14. **if** (**not** ok) **then**
  15. write (‘La suma no quedó entre 10 y 30’);
  16. **end**.

1. 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****: Realice 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*.

1. **program** programadores;
2. **procedure** leerDatos(**var** legajo: integer; salario : real);
3. **begin**
4. writeln(‘Ingrese el nro de legajo y el salario”);
5. read(legajo);
6. read(salario);
7. **end**;
8. **procedure** actualizarMaximo(nuevoLegajo:integer; nuevoSalario:real; **var** maxLegajo:integer);
9. **var**
10. maxSalario : real;
11. **begin**
12. **if** (nuevoLegajo > maxLegajo) **then begin**
13. maxLegajo:= nuevoLegajo;
14. maxSalario := nuevoSalario
15. **end**;
16. **end**;
17. **var**
18. legajo, maxLegajo, i : integer;
19. salario, maxSalario : real;
20. **begin**
21. sumaSalarios := 0;
22. **for** i := 1 **to** 130 **do begin**
23. leerDatos(salario, legajo);
24. actualizarMaximo(legajo, salario, maxLegajo);
25. sumaSalarios := sumaSalarios + salario;
26. **end**;
27. writeln(‘En todo el mes se gastan ‘, sumaSalarios, ‘ pesos’);
28. writeln(‘El salario del empleado más nuevo es ‘,maxSalario);
29. **end**.
30. **a.** Realizar un módulo que reciba un par de números (numA,numB) y retorne si numB es el doble de numA.
    1. 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).*

1. 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.
2. **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.
   1. 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.
3. 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.
4. 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 nro de inscripción par.
5. 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.

1. 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.

1. **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 ó 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$S320 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.

# EJERCICIOS ADICIONALES

**1.** Dado el siguiente programa:

**program** Ejercicio1\_ad;

**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**.

1. ¿Qué imprime si se leen los valores num1=10 y num2=5 ?
2. ¿Qué imprime si se leen los valores num1=5 y num2=10 ?
3. Realice 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)*

1. 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.
2. **a.** Realizar un módulo que reciba como parámetro el radio de un círculo y retorne su diámetro y su perímetro.
   1. 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).
3. 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.** Analice sus soluciones a dichos problemas, e identifique:

* + - ¿Qué porciones de su código podrían modularizarse? ¿En qué casos propondría una estructura de módulos anidada?
    - ¿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?
    - ¿Qué mecanismos de comunicación conviene utilizar entre los módulos propuestos?
  1. Implemente nuevamente los 3 programas, teniendo en cuenta los módulos propuestos en el inciso anterior.