



Trabajo Práctico Nº 6

Funciones y Procedimientos

Observación general: Si el ejercicio pide explícitamente una función debe realizar una función no un procedimiento (de igual manera para el caso cuando se pide un procedimiento). Si no se especifica cómo implementar la primitiva, entonces puede elegir la forma que le parezca más conveniente (función o procedimiento) siguiendo los criterios que fueron indicados en clase.

Ejercicio 1: Considere la siguiente definición de la función Invertir:

```
function Invertir( num: integer ):integer;
    {Objetivo: Invierte el orden de los dígitos del número entero Num}
var
    inv: integer;
begin
    inv := 0;
    while num > 0 do
        begin
            inv := inv * 10 + (num mod 10);
            num := num div 10;
        end;
    Invertir := inv;
end;
```

Realice un programa que, utilizando la función dada, determine si un número natural **Num** ingresado por el usuario es o no capicúa.

Por ejemplo: si Num = 12321, el programa deberá mostrar por pantalla "El número 12321 ES CAPICUA".

si Num = 2343, el programa deberá mostrar por pantalla "El número 2343 NO ES CAPICUA".

Ejercicio 2:

Implemente las siguientes funciones, indicando casos de prueba para la verificación de cada una de ellas:

a) **function** potencia(base, exponente: integer): integer; { Calcula $Base^{Exponente}$ }

b) **function** primo(nro: integer):boolean; { Determina si Nro es primo }

c) **function** cantVeces(d, Nro: integer): integer; { Calcula la cantidad de veces que aparece el dígito d en el número Nro }

d) **function** esBisiesto(a: integer): boolean; { Determina si el año a es bisiesto }

Ejercicio 3:

Escriba una función para determinar si una fecha es válida. La fecha es representada por tres números enteros día, mes y año (utilice la sentencia CASE). Luego, escriba un programa que solicite al usuario una fecha e indique por pantalla si la misma es válida o no



Ejercicio 4: Utilizando las funciones definidas en el ejercicio 2, resuelva los siguientes incisos:

- Escriba un **programa** que dado un archivo de números entero "numeros.dat", cree otro archivo "numeros2.dat" conteniendo sólo los números **primos o capicúas** del primero.
- Escriba un **programa** que determine si un número N es o no **especial**. Un número N es **especial** si cada dígito que está en N aparece tantas veces como su propia magnitud. Por ejemplo, si el dígito 3 aparece en un número entonces deberá aparecer en exactamente 3 posiciones distintas para que dicho número sea especial; si el 1 aparece, deberá aparecer sólo una vez; si el 2 aparece está dos veces, etc. Por ejemplo, los números 333, 3313, 4234132434 son números especiales.
- Escriba un **programa** para mostrar por pantalla todos los números primos entre 1 y un valor natural ingresado por el usuario (inclusive).

Por ejemplo: Ingrese un valor: 32

 Los números primos entre 1 y 32 son: 2 3 5 7 11 13 17 19 23 29 31

Ejercicio 5: Conceptos

- Explique las diferencias y similitudes entre procedimientos y funciones en Pascal. ¿Cuándo usaría cada uno?
- Explique las diferencias entre parámetros por valor y por referencia. Indique además que es lo que se permite para cada uno de ellos en los parámetros efectivos.
- Explique las diferencias entre las siguientes clases de errores: error de compilación, error de ejecución y error lógico.

Ejercicio 6: Considere el programa "Ejercicio". Escriba otro código fuente que tenga el mismo efecto al dado (para cualquier valor de la variable entrada) pero reemplazando la sentencia CASE por sentencias IF-THEN-ELSE.

```

program Ejercicio;
var opcion:integer;
begin
  writeln('ingrese opción ');  readln(opcion);
  case opcion of
    1..9: write('primera');
    10,11: write('segunda');
    20..30,41,50..60: write('tercera');
    else write('cuarta');
  end;
  writeln('Final');
end.
  
```

Ejercicio 7: Encuentre los errores sintácticos en cada una de las siguientes secuencias de instrucciones. Asuma que todas las variables utilizadas han sido declaradas e inicializadas con valores válidos.

<pre> a) puntos := 0; Case cuenta of 1, 2 : puntos := puntos + 1; 3..8 : puntos := puntos + 2; 8: puntos := puntos + 4 end; </pre>	<pre> b) case (p + q) of 1.84 : write (p); 4.24 : write (q) end; </pre>	<pre> c) case a > b of 'true' : begin c := b - a; b := b + 1.0 end 'false' : c := a - b; end; </pre>
---	--	--

Ejercicio 8: Implemente un procedimiento en PASCAL para mostrar por pantalla la calificación de un alumno en función de la nota obtenida. El procedimiento recibirá por parámetro una nota numérica correspondiente a un entero entre 0 y 10. La calificación se determina de acuerdo al siguiente detalle:

De 0 a 3: **Aplazado**; de 4 a 6: **Aprobado**; de 7 a 8: **Distinguido**; 9: **Excelente**, y 10: **Sobresaliente**.



Ejercicio 9: Escriba un programa en Pascal que procese una secuencia de caracteres ingresada por teclado y terminada en punto, codificándola de la siguiente manera: cada vocal se reemplaza por el carácter que se indica en la tabla que sigue, el resto de los caracteres se mantienen sin cambios.

A, a	E, e	I, i	O, o	U, u
@	#	\$	%	&

(Implemente una función que reciba una vocal y retorne la codificación correspondiente. Utilice la sentencia CASE para la transformación.)

Por ejemplo, si el usuario ingresa: **Ayer, lunes, salimos a las once y 10.**

La salida del programa debería ser: **@y#r, l&n#s, s@l\$m%\$ @ l@s %nc# y 10.**

Ejercicio 10: Indique cuántos parámetros por valor y cuántos por referencia hay en cada uno de los siguientes procedimientos y funciones:

1. **PROCEDURE Eje1** (**var** letra1,letra2:char; N1,N2:integer; **var** Error:boolean);
2. **PROCEDURE Eje2** (**var** A:char; **var** b:integer; **var** c:boolean);
3. **FUNCTION F1** (a,b:integer; es: boolean):real;
4. **FUNCTION LeeLetra**: CHAR;
5. **FUNCTION LeeNumero** (l:char; **var** error:boolean):integer;

Ejercicio 11: Para cada uno de los siguientes incisos realice un programa en PASCAL implementando funciones o procedimientos de forma de obtener una correcta división del problema.

Suponga que cuenta con un archivo A de números enteros.

- a) Generar otro archivo B con los elementos de A que sean capicúas y tengan una cantidad impar de dígitos y finalmente mostrar todos los elementos del archivo B.
- b) Generar otro archivo B con los elementos de A que sean primos o tengan todos los dígitos impares y finalmente mostrar todos los elementos del archivo B.

Ejercicio 12: Conceptos

- a) Explique cuándo dos tipos son idénticos, cuando compatibles y cuando se cumple la compatibilidad de asignación. Escriba 2 ejemplos de cada uno.
- b) Explique por qué en Pascal se requiere declarar un identificador de tipo, cuando un procedimiento o una función tienen un archivo como parámetro.
- c) Indique las diferentes opciones que existen para los parámetros efectivos cuando se corresponden con un parámetro formal por valor o por un parámetro formal por referencia y las compatibilidades de tipo necesarias.
- d) Analizar cuáles de las invocaciones a procedimientos o funciones detalladas a continuación son correctas:

VAR w: Char; x: Integer; y: Real; z: Boolean;

PROCEDURE Proc1 (a,b: Integer; **var** c: Char);

FUNCTION Funcion1 (x: char): Real;

FUNCTION Funcion2 (**VAR** a: Real; b: Boolean): Integer;

BEGIN ... END;

BEGIN ... END;

BEGIN ... END;

1. Proc1(7, y, w);

2. Proc1(7, y, c);

3. Proc1(27, x, w, w);

4. Proc1(2.4, 5+8, w);

5. Proc1(7, 5, 'c');

6. x:= Funcion1(w);

7. y:= Funcion1(w);

8. y:= Funcion1('x');

9. Funcion1(w);

10. x := Funcion2(y, false);

11. y := Funcion2(y, true);

12. x := Funcion2(3+5, z);

13. x := Funcion2(3.5+y, z);



Ejercicio 13:

- a) Escriba un procedimiento *diaAnterior* (*var d, m, a: integer*) que reciba una fecha representada a través de tres enteros *d, m* y *a* y retorne fecha anterior. Puede asumir que *d, m* y *a* representan una fecha válida.
- b) Realice un programa que solicite al usuario una fecha y muestre por pantalla la fecha anterior que resulta de utilizar el procedimiento del inciso a).
Realice trazas para los valores $d=5, m=10, a=2012$ y para $d=1, m=3, a=2004$

Ejercicio 14: Escriba un procedimiento en Pascal que muestre los divisores de un número *n* recibido como parámetro. Escriba un programa que lea un archivo de números enteros "datos.dat" y utilizando el procedimiento definido antes muestre por pantalla los divisores de cada número del archivo.

Ejercicio 15: Implemente un procedimiento que dado un dígito $d \in [1..9]$ muestre por pantalla el siguiente renglón:

1 2 3 .. d .. 3 2 1

Por ejemplo, si $d = 6$ el procedimiento deberá imprimir

1 2 3 4 5 6 5 4 3 2 1

El encabezamiento del procedimiento debe ser:

PROCEDURE ImprimeRenglón (dígito: integer);

Escriba un programa en Pascal utilizando dicho procedimiento, para que solicite un dígito *d* al usuario, y muestre por pantalla una figura como la siguiente:

```

1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
...
1 2 3 .....d..... 3 2 1

```

Ejercicio 16: Escriba una función para que dado un archivo de números enteros y un número *N*, indique si el número *N* pertenece al archivo. Luego, con éste procedimiento y dos archivos de números, *A.dat* y *B.dat*, escribir un programa para mostrar por pantalla los elementos comunes a ambos archivos.

(Este ejercicio podrá ser entregado al asistente para su corrección).