



Trabajo Práctico 6

Funciones, Procedimientos y División de Problemas

Ejercicio 1: Considere definida la función Invertir.

```
function Invertir( num: integer ):integer;
{ Objetivo: Invierte el orden de los dígitos del número entero
  Num: número que se desea invertir.
  Salida: Devolverá un entero con los dígitos de Num en orden inverso }
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

function factorial(Nro: integer): integer; { Calcula el factorial de un número Nro }

function potencia(Base, Exponente: integer): integer; { Calcula $Base^{Exponente}$ }

function primo(Nro: integer):boolean; { Determina si Nro es primo }.

function techo(Nro: real): integer; { Calcula $\lceil Nro \rceil$ }

{La función techo se aplica a un número real Nro y devuelve el mínimo número entero k no inferior a Nro}

Ejercicio 3: Escriba un programa en Pascal que lea un par de números n y e, y muestre por pantalla todos los números primos comprendidos entre 1 y n^e (n elevado a la e). Nota: puede utilizar primitivas de ejercicios anteriores.

Por ejemplo: para $n = 2$ y $e=5$ (donde n^e es 32), el programa deberá mostrar por pantalla:

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

Ejercicio 4: Escriba un programa en Pascal dividiendo el problema en subproblemas, que elimine de un archivo de enteros todas las componentes que sean **primos** o sean **capicúas**.



Ejercicio 5: Suponga que cuenta con un archivo A de números enteros ya ingresados. Para cada uno de los siguientes incisos se desea generar otro archivo nuevo con los elementos de A que cumplan lo indicado:

- a) Sean capicúas y tengan una cantidad impar de dígitos.
- b) Sean primos o tengan todos los dígitos impares.
- c) Tengan una cantidad par de dígitos, no sean capicúas y tengan al menos un dígito par.

Para cada uno de los incisos anteriores (en forma individual) se solicita que:

- i) Divida el problema en subproblemas y haga un gráfico o esquema de su propuesta de diseño para la solución.
- ii) Describa las funciones y procedimientos necesarios identificando los parámetros de entrada y salida, agregando una breve descripción del objetivo de la primitiva.
- iii) Realice un programa en PASCAL que resuelva el problema. No es necesario implementar las primitivas, simplemente deberá declarar los encabezados de cada una.

Ejercicio 6: Fecha Válida

Realice una función para determinar si una fecha es válida. La fecha es representada por tres números enteros **dia**, **mes** y **año** ingresados por el usuario. Por ejemplo si dia= 21, mes= 10 y año=2008, la fecha es válida. Si dia=29, mes=2 y año = 2010 la fecha no es válida. Utilice para determinar la validez de la fecha la sentencia CASE.

Ejercicio 7: Codificador

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 (incluyendo a las vocales acentuadas) se mantienen sin cambios.

a	e	i	o	U
@	#	\$	%	*

Realice 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 8: Explique la diferencia que hay entre parámetros por valor y por referencia en Pascal. Indique cuantos parámetros por valor y cuantos 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 9: (a) 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. (b) Analizar cuáles de las invocaciones a procedimientos o funciones detalladas a continuación son correctas en base a lo indicado en el inciso (a) y a las siguientes declaraciones:

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

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

FUNCTION Funcion1 (x: char):Real;
BEGIN ... END;

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

- | | | |
|------------------------|-----------------------|------------------------------|
| 1. Proc1(7, y, w); | 6. x:= Funcion1(w); | 10. x := Funcion2(y, false); |
| 2. Proc1(7, y, c); | 7. y:= Funcion1(w); | 11. y := Funcion2(y, true); |
| 3. Proc1(27, x, w, w); | 8. y:= Funcion1('x'); | 12. x := Funcion2(3+5, z); |
| 4. Proc1(2.4, 5+8, w); | 9. Funcion1(w); | 13. x := Funcion2(3.5+y, z); |
| 5. Proc1(7, 5, 'c'); | | |

Ejercicio 10: Operaciones

Realice un programa en PASCAL que lea un número real A, un operador aritmético OP (que puede ser +, -, *, /) y otro número real B y calcule el valor de la expresión A OP B. Implemente una función que evalúe la expresión.

Un ejemplo de la interacción del programa con el usuario es:

Ingrese A: 2.0 <Enter>
Ingrese OP: + <Enter>
Ingrese B: 4.5 <Enter>

El resultado es: 6.5

Otro ejemplo es:

Ingrese A: 2.0 <Enter>
Ingrese OP: / <Enter>
Ingrese B: 0 <Enter>

No se puede calcular (división por cero).

Y uno más:

Ingrese A: 2.0 <Enter>
Ingrese OP: - <Enter>
Ingrese B: 4.5 <Enter>

El resultado es: -2.5



Ejercicio 11: 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 sería:

PROCEDURE ImprimeRenglón(digito: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 12: Se dice que M es el número maximal para N , si M es el **mayor número que puede formarse usando los dígitos de N** . Ejemplos: Si $N=125345$, el número maximal M es 554321; si $N=2756$, M es 7652.

Escriba un programa en Pascal que lea dos números naturales a y b y muestre por pantalla todos los números Num comprendidos entre a y b que verifiquen que coinciden con su maximal:

Por ejemplo: para $a = 320$ y $b = 332$, el programa deberá mostrar por pantalla:
Los números entre 320 y 332 que coinciden con su maximal son: 320, 321, 322, 330,331, 332

Ejercicio 13: Conteste las siguientes preguntas dando un ejemplo en el caso que la situación planteada sea posible, o fundamentando su respuesta con conceptos teóricos. Dentro de un programa en Pascal:

- ☞ ¿Pueden dos procedimientos tener el mismo nombre?
- ☞ ¿Pueden haber dos funciones con el mismo identificador?
- ☞ ¿Puede un identificador de constante ser igual a un identificador de variable?
- ☞ Indique cuando un procedimiento P puede llamar a una función F que está declarada dentro de otro procedimiento Q , y cuando no.
- ☞ ¿Puede una variable local tener como nombre V si está declarada dentro de un procedimiento cuyo nombre también es V ?
- ☞ ¿Puede una variable local tener como nombre V si está declarada dentro de una función cuyo nombre es V ?
¿Hay alguna diferencia con respecto a que V sea un procedimiento?

Ejercicio 14: Reescriba el programa generado en el **ejercicio 1** utilizando el siguiente procedimiento.

```
procedure Invertir( num: integer; var inv:integer );
{ Objetivo: Invierte el orden de los dígitos del número entero
  num: número que se desea invertir.
  Salida: Devolverá un entero con los dígitos de Num en orden inverso }
begin
  inv := 0;
  while num > 0 do
  begin
    inv := inv * 10 + (num mod 10);
    num := num div 10;
  end;
end;
```



Ejercicio 15: Suponga que cuenta con tres archivos A, B y C, y todos tienen ingresados números reales. Para cada uno de los siguientes incisos se desea generar otro archivo nuevo con aquellos elementos de A, B y C, que respeten lo indicado:

- Copiar todos los elementos que se encuentre en A y en B pero no en C.
- Copiar aquellos elementos que aparezcan en A una cantidad par de veces o aparezcan en B una cantidad impar de veces, y que si aparece en C entonces no debería aparecer en la misma cantidad que apareció en A o en B.
- Copiar todo elemento que ocurra en A antes que en B, y en B antes que en C. Si el elemento no aparece en algún archivo entonces no se copia.

Para cada uno de los incisos anteriores (en forma individual) se solicita que:

- Divida el problema en subproblemas y haga un gráfico o esquema de su propuesta de diseño para la solución.
- Describa las funciones y procedimientos necesarios identificando los parámetros de entrada y salida, agregando una breve descripción del objetivo de la primitiva.
- Realice un programa en PASCAL que resuelva el problema. No es necesario implementar las primitivas, simplemente deberá declarar los encabezados de cada una.

Ejercicio 16: Considere las siguientes sumatorias.

a) $1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!}$

b) $n + \frac{n-1}{2!} + \frac{n-2}{3!} + \frac{n-3}{4!} + \dots + \frac{1}{n!}$

c) $\frac{1}{n!} + \frac{2^2}{(n-1)!} + \frac{3^4}{(n-2)!} + \frac{4^6}{(n-3)!} + \frac{5^8}{(n-4)!} + \dots + \frac{n^{2(n-1)}}{1!}$

- Calcule el resultado, para cada uno de los incisos, considerando $n=0$, $n=2$ y $n=6$ (suponiendo $m=4$).
- Para cada inciso realice un programa que calcule el resultado de dicha sumatoria. *Divida el problema en partes considerando que las partes de una sumatoria se denominan sumandos y que las partes de un producto se conocen como factores. Así podría definirse sumando(i) o termino(i) que retorne el valor del término de dicha posición, que a su vez puede calcularse en función de factor(i), etc.*

Ejercicio 17: El período de un caracter se define como la distancia entre dos apariciones del mismo. Por ejemplo, si tiene la siguiente secuencia “parte de la traza es trabajosa.” El período de **t** es 8 que es la cantidad de caracteres que separan las t. Cuando se considera el caracter **a** se observa que existen varias ocurrencias con diferentes períodos 8, 3, 1, 6, 1 y 3 (obsérvese que sólo mide la distancia de un caracter al próximo inmediato idéntico, pero no se mide la distancia entre cualquier par, por ejemplo, la distancia entre la **a** de **Parte** y la **a** de **traza** no se mide). Diremos que un **caracter es periódico** si ocurre siempre con la misma periodicidad.

Considere que se dispone de dos archivos “valores.in” y “registro.dat”, donde “valores.in” registra que elementos aparecen en “registro.dat”. Por ejemplo, para la secuencia de arriba el archivo “valores.in” contendría los siguientes valores “p a r t e d l z s b j o .”.

- Realice un programa que muestre todos los caracteres periódicos (junto con su periodicidad) a basado en los datos de “registro.dat” pudiendo ayudarse del archivo “valores.in”.
- Realice un procedimiento que reciba como entrada un archivo que contiene una secuencia de caracteres y genere otro archivo “valores.in” que contenga únicamente los caracteres que aparezcan en el archivo provisto, con la particularidad que cada carácter aparecerá solo una vez. Por ejemplo, el carácter **a** aparecerá solo una vez en “valores.in” a pesar que en la secuencia aparece reiteradas veces y el carácter **u** no aparecerá ya que no se encuentra en la secuencia ingresada



Ejercicio 18: Suponga que cuenta con un archivo "alumnosAM2.dat" de números enteros que contiene el número de registro de todo alumno inscripto para cursar la materia Análisis Matemático 2. También se dispone de 3 archivos adicionales con las notas que obtuvieron dichos alumnos en cada examen. Llamaremos "parcial1.dat" y "parcial2.dat" a los archivos que contienen las notas de los parciales y "recu.dat" al archivo que contiene la nota del recuperatorio. Cada archivo de notas almacenará la información según el siguiente formato "registro nota registro nota ... <EOF>". Las notas solo podrán ser valores enteros entre cero y diez, y todo registro que aparece en alguno de los archivos de notas deberá estar presente en el archivo de alumnos (*estas propiedades no necesitan ser verificadas*). Si un alumno no se presentara a rendir un parcial, su nro de registro no aparecerá en el archivo correspondiente, por ejemplo, si el alumno 96546 no viene al 2^{do} parcial, dicho registro no aparecerá en el archivo "parcial2.dat". Se considera aprobado un examen si ha obtenido una nota mayor o igual a 6. Ninguno de los archivos se encuentra ordenado.

Se solicita generar tres archivos donde uno de ellos contenga los registros de todos los alumnos que cursaron la materia (es decir, sacaron más de 6 en ambos parciales o aprobaron el recu) y el promedio obtenido entre todas sus notas, otro archivo que contenga los registros de aquellos alumnos que perdieron la materia y otro archivo con los números de registro de los alumnos que estuvieron ausentes a todos los parciales.

- i) Divida el problema en subproblemas y haga un gráfico o esquema de su propuesta de diseño para la solución.
- ii) Describa las funciones y procedimientos necesarios identificando los parámetros de entrada y salida, agregando una breve descripción del objetivo de la primitiva.
- iii) Realice un programa en PASCAL que resuelva el problema. No es necesario implementar las primitivas, simplemente deberá declarar los encabezados de cada una.



Ejercicio 19: Considere el siguiente programa en Pascal y muestre que salida se producirá en la pantalla como resultado de su ejecución.

```
PROGRAM p19;
procedure multiplicarXvalor(x,y,z: integer); // Usando el método de sumas sucesivas
var i:integer;
begin
    z := 0;
    for i:= 1 to y do z := z + x;
end;
procedure multiplicarXref(x,y: integer; var z: integer); // Usando el método Ruso
begin
    z := 0;
    while x >0 do begin
        if (x mod 2) <> 0 then z := z + y;
        x := x div 2;
        y := y + y;
    end;
end;
procedure multiplicarXref2(var x,y,z: integer); // Usando sumas sucesivas con efectos colaterales
begin
    while y > 0 do begin
        z := z + x;
        y := y - 1;
    end;
end;
var a,b,z: integer;
begin
    a:=12; b:=4; z:=1; writeln('1- A:',a, ' B:',b, ' Z:',z);
    multiplicarXvalor(a,b,z);
    writeln('2- A:',a, ' B:',b, ' Z:',z); z := 1;
    multiplicarXref(a,b,z);
    writeln('3- A:',a, ' B:',b, ' Z:',z); z := 1;
    multiplicarXref(a,b,z);
    writeln('4- A:',a, ' B:',b, ' Z:',z); z := 1;
    multiplicarXref2(a,b,z);
    writeln('5- A:',a, ' B:',b, ' Z:',z); z := 1;
    multiplicarXref2(a,b,z);
    writeln('6- A:',a, ' B:',b, ' Z:',z);
    readln;
end.
```