



Trabajo Práctico 5

Funciones y Procedimientos. Archivos de Texto.

Estrategias de resolución de problemas basadas en el uso de primitivas.

Versión 2

Ejercicio 1: Realice un programa determine si un número ingresado por el usuario es o no capicúa utilizando la función Invertir. 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"

```
function Invertir( num: integer ):integer;
{Objetivo: Invierte el orden de los dígitos de un número entero
Entrada: el parámetro "Num" recibe el número que se desea invertir.
Salida: Retornará un entero con los dígitos de "Num" en orden inverso
Ejemplo: si "num" es 1234 retornará 4321}
var
    inv, ultimo: integer;
begin
    inv := 0; {variable auxiliar en la que se construye el "invertido"}
    while num > 0 do
    begin {voy sacando dígitos de Num del menos al más significativo}
        ultimo := num mod 10;
        num := num div 10;
        inv := inv * 10 + ultimo; {voy agregando al final del invertido}
    end;
    Invertir := inv;
end;
```

Ejercicio 2: Implemente las siguientes funciones, indicando casos de prueba la verificación de cada una de ellas:

- 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 mayor o igual a Nro
ejemplos: techo(2.4) es 3 y techo(2.0) es 2}
- function** fechaValida(dia,mes,anio:integer):boolean;
{La fecha es representada por tres números enteros día, mes y año. 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.}

Observación general: A lo largo de todo el práctico, si un ejercicio indica implementar una función o procedimiento pero ese ejercicio no requiere realizar un programa para alguna tarea específica, entonces debe implementar un programa de prueba que llame a la función o procedimiento realizado.

Ejercicio 3: Escriba un programa en Pascal que utilice las primitivas del ejercicio 2 y muestre por pantalla todos los números primos entre 1 y un valor natural ingresado por el usuario (inclusive).

Ingrese un valor: 32

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 que elimine de un archivo de enteros todos los elementos que sean **primos o capicúas**. Haga una división del problema en subproblemas, y utilice o construya primitivas adecuadas.



Ejercicio 5: 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 de un número entero
Entrada: el parámetro "Num" recibe el número que se desea invertir.
Salida: Retornará un entero con los dígitos de "Num" en orden inverso
Ejemplo: si "num" es 1234 retornará 4321}
begin
  inv := 0;
  while num > 0 do
    begin
      inv := inv * 10 + (num mod 10);
      num := num div 10;
    end;
end;
```

Ejercicio 6: Preguntas Teóricas

- 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.

Ejercicio 7: Para cada uno de los siguientes incisos realice un programa en PASCAL implementando funciones o procedimientos según su criterio, de forma de obtener una correcta división del problema en sub-problemas, y realice además un esquema que ilustre su propuesta de diseño para la solución. Suponga que cuenta con un archivo A de números enteros. Indique los casos de prueba que usó.

- 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.
- 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.
- Generar otro archivo B con los elementos de A que tengan una cantidad par de dígitos, no sean capicúas y tengan al menos un dígito par y finalmente, mostrar todos los elementos del archivo A y luego los del archivo B.

Observación general: A lo largo de todo el práctico, si el ejercicio pide explícitamente una función debe realizar una función (de igual manera para procedimientos). Sin embargo, si no se especifica como implementar la primitiva, entonces debe elegir la mejor manera (función o procedimiento) siguiendo los criterios que fueron indicados en clase.

Ejercicio 8: Una forma de clasificar los sonidos del habla y en particular de las consonantes de un idioma es según el punto de articulación (otras formas son la utilización de la corriente de aire producida por los mecanismos subglóticos, el estado de la glotis y la configuración de la cavidad oral y nasal). Utilizaremos la siguiente clasificación que es una mezcla entre el punto y modo de articulación:

- Dentales (punto de articulación en los dientes): consonantes d, t, s, z
- Labiales (punto de articulación en los labios): consonantes b, p, f, v
- Guturales (punto de articulación en la garganta): consonantes c, k, q, g, x
- Nasales (la corrientes de aire espirado sale por la nariz): consonantes m, n
- Líquidas (el aire se desliza por los lados): consonantes l, r

- Realice una primitiva que dado un carácter retorne su clase, representada por un valor entero (1: dental, 2: labial, 3: gutural, 4: nasal, y 5: líquida), en el caso que el carácter no se encuentre retornará un 0.
- Realice una primitiva que dado un carácter muestre por pantalla su clasificación (sugerimos usar CASE).
- Realice un programa que solicite al usuario una palabra (terminada en ".") y por cada carácter ingresado, si el mismo aparece en la clasificación mostrar la clase por pantalla y si no aparece no mostrar nada.
- Realice un programa que reciba un archivo de texto y que cuente cuantas consonantes hay de cada clase.



Ejercicio 9: 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 10: Preguntas Teóricas

- a) Indique similitudes y diferencias entre un archivo de tipo predefinido TEXT y uno declarado como FILE OF CHAR.
- b) Explique cuando dos tipos son idénticos, cuando compatibles y cuando se cumple la compatibilidad de asignación. Escriba 2 ejemplos de cada uno.
- c) Explique por qué en Pascal se requiere declarar un identificador de tipo, cuando un procedimiento o una función tienen un archivo como parámetro.

Ejercicio 11:

(a) Escriba un procedimiento que indique cuantas líneas tiene un archivo de texto.

(b) Escriba una función que busque cuantas veces está un carácter E (ingresado por el usuario) en cada línea de un archivo de texto.

Ejercicio 12: Considere que dispone de un archivo de texto donde cada línea contiene una fecha según el siguiente formato DD/MM/AAAA. Donde DD representa que son dos dígitos para el día, MM representa dos dígitos para el mes y AAAA representa que el año son cuatro dígitos. Haciendo una división del problema en sub-problemas, escriba primitivas adecuadas y un programa que permita mostrar por pantalla todas las fechas inválidas que encuentre en el archivo, indicando la línea donde se encuentra presente cada fecha inválida. Indique casos de prueba.

Ejercicio 13:

- (a) Escriba un procedimiento que agregue dos líneas en blanco a un archivo de texto. ¿Podría utilizarse una función para dicha tarea? Justifique su respuesta.
- (b) Escriba un procedimiento o función que reemplace un carácter E por otro C (ingresados por el usuario) en un archivo de texto, e indique en cuantas líneas lo reemplazó. Indique los casos de prueba.

Ejercicio 14: Suponga que cuenta con tres archivos de texto A, B y C los cuales tienen una secuencia de 0 o más números reales. Para cada uno de los siguientes incisos se desea generar otro archivo de texto con aquellos elementos de A, B y C, que respeten lo indicado:

- a) Copiar todos los elementos que se encuentre en A y en B pero no en C.
- b) Copiar aquellos elementos que aparezcan en A, B o C. No deberán copiarse aquellos que aparezcan en al menos dos archivos en la misma posición.
- c) 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.

Realice para cada inciso un programa en PASCAL que resuelva el problema y divida el problema en sub-problemas haciendo 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. Indique los casos de prueba de cada inciso.



Ejercicio 15:

- (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 y las compatibilidades de tipo necesarias.
- (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 16: Preguntas Teóricas

- a) Explique cuando un identificador es visible en un bloque. Ejemplifique en Pascal.
- b) Indique la definición de entorno de referencia de un bloque de Pascal. Realice un ejemplo donde se vea la diferencia entre entorno local, no local y global.

Ejercicio 17: En una competencia por internet fueron seleccionados de todo el mundo un grupo de participantes que en la ronda semi-final tuvieron que demostrar sus habilidades en distintos juegos. En el archivo "participantes.semi" se encuentran los números de usuario de todos los participantes de la ronda semi-final (números naturales). En los archivos "juego.uno", "juego.dos" y "juego.tres", se encuentran los resultados de cada uno de los juegos. Cada uno de estos tres archivos tiene un par de elementos que corresponde al número de usuario de un participante, seguido del puntaje obtenido en ese juego (número natural), y así para todos los que participaron en ese juego. En cada uno de los siguientes incisos ponga especial atención en dividir el problema en partes y crear primitivas que faciliten la reutilización de código. Indique ejemplos significativos.

- a) Se debe escribir un programa en Pascal que dado un número de usuario indique si puede participar en la final de la competencia. Un participante podrá participar en la final si se encuentra en los tres archivos de puntajes.
- b) Se debe escribir un programa en Pascal que permita mostrar quien fue el mejor participante de cada juego. La mejor participación en un juego consiste de haber obtenido el puntaje más alto.
- c) Se debe escribir un programa en Pascal que permita mostrar quien fue el mejor participante de cada juego y que puntajes obtuvo ese mismo participante en los otros dos juegos.
- d) Se debe escribir un programa en Pascal que permita crear un archivo de enteros con los participantes que obtuvieron más de una cantidad "tope" de puntos en los tres juegos (tope ingresado por el usuario).



Ejercicio 18: 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. Este tipo de procesamiento es una versión modesta de encriptación de datos.

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 19: *En este ejercicio preste especial atención a la correcta **división del problema**. (Parcial 2013)*

Considere que una empresa de transportes de latas de bebidas tiene dos camiones y en cada uno de ellos puede llevar hasta 100 cajas de latas. Cada caja tiene un código individual único representado por un número entero el cual se utiliza para su seguimiento durante el transporte. La empresa tiene dos archivos de enteros "camión1" y "camión2", cada archivo con los códigos de las cajas que ya han sido cargadas en cada camión y que aún no ha salido de viaje. Además, para el nuevo lote de cajas que no fue cargado aún, la empresa tiene otro archivo de enteros llamado "lote_a_cargar" con los códigos de todas las cajas de ese lote.

Se debe escribir un programa en Pascal que leyendo la información de los archivos indicados antes, permita al operador del sistema realizar estas tareas: (1) mostrar cuántas cajas pueden aún cargarse en cada camión, (2) mostrar si dado el espacio disponible entre los dos camiones, alcanza para cargar todas las cajas de "lote_a_cargar", o de lo contrario cuántas cajas quedarían sin cargar, y (3) dado un código, mostrar en donde está esa caja (camión 1, camión 2, lote, o no está en transporte)

Por ejemplo, si se tienen estos datos,

Camión1	201 202 203 204 205 206
Camión 2	301 302 303 304 305
Lote_a_cargar	91 92 93 94 95 96 97

Entonces se mostrará por pantalla:

Ingrese opción (1) disponible (2) hay espacio (3) donde caja > 1
Pueden cargarse. Camión 1: 94 cajas, y camión 2: 95 cajas
Ingrese opción (1) disponible (2) hay espacio (3) donde caja > 2
El lote puede cargarse completamente
Ingrese opción (1) disponible (2) hay espacio (3) donde caja > 3
Ingrese código de caja a buscar: 94
La caja buscada está en el lote a cargar

Ejercicio 20: 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.

- a) Realice una función que determine si un número es maximal.
- b) Realice una función que dado un número retorne el maximal que puede formarse utilizando sus dígitos.
- c) 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 21: 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 22: Escriba un programa que permita ingresar la fecha del día actual, y luego solicite al usuario las fechas de ida y vuelta de un viaje. El ingreso de cada fecha deberá realizarlo permitiendo el ingreso de una secuencia de caracteres con el formato dd/mm/aa. Controle que todas las fechas sean válidas y que caso que no lo sean continúe solicitando la fecha hasta que ingrese una fecha válida, a tal efecto utilice la función implementada en el ejercicio 10. Luego controle que la fecha de viaje sea posterior o igual a la del día actual, y que la de regreso sea posterior o igual a la de ida del viaje. Finalmente si pasa todos los controles solicite el número de tarjeta de crédito (que consiste de 4 números de exactamente 4 cifras) y la fecha de vencimiento de la tarjeta (la cual debe estar vigente), sobre dichos valores realice los controles necesarios. Indique ejemplos significativos para los casos de prueba.

Ejercicio 23: Escriba un programa que lea un archivo de texto con pares LU Nota (donde Nota es A B C o D) y genere un nuevo archivo de texto que tenga un listado para imprimir con un alumno por línea con el formato LU <resultado> (EOL). donde <resultado> es Aprobado si Nota es A o B y Desaprobado si es C o D. Indique ejemplos significativos para los casos de prueba.

Ejercicio 24: Suponga que dispone de un archivo de texto llamado "impresora.conf" que contiene las preferencias de una impresora. Por ejemplo, el contenido del archivo podría ser el siguiente

```
nombre impresora = LASER P-PITA
tamaño de hoja = A4
calidad = borrador
orientación = apaisada
predeterminada = si
```

En este ejemplo, cada línea corresponde a un par (preferencia, valor) donde el nombre de la preferencia se encuentra a la izquierda del signo igual y el valor se encuentra a su derecha.

Escriba un programa que muestre el contenido del archivo, numerando cada una de las preferencias y permita seleccionar al usuario (mediante un número) la preferencia a modificar. Luego de seleccionar la preferencia, se debe permitir que el usuario ingrese el nuevo valor y este deberá reflejarse modificando el archivo. Deberán existir dos opciones adicionales, cero en el caso de no querer modificar nada y menos uno (-1) si se desea agregar una nueva preferencia (no es necesario verificar que no sea un duplicado). Indique otros ejemplos significativos para los casos de prueba (diferentes al dado).



Ejercicio 25 (importante: se evaluará la correcta división del problema en subproblemas y la elección adecuada de procedimientos o funciones para implementarlos).

En una biblioteca cada libro tiene un código único que está representado por un número entero. El sistema de la biblioteca dispone de 3 archivos de números enteros (ordenados de menor a mayor): el archivo "disponibles" con los códigos de los libros que están en la biblioteca y pueden ser prestados; el archivo "prestados" que tiene los códigos de los libros que fueron prestados y no están en la biblioteca; y el archivo "reservados" con los libros que fueron prestados, pero que además fueron reservados por alguien para sacarlos cuando sean devueltos.

Además, existe otro archivo llamado "pedidos" que tiene los códigos de los libros que quieren ser retirados de la biblioteca. Observe que puede haber archivos vacíos.

Escriba un programa en Pascal que lea del archivo "pedidos" y para cada código N encontrado, escriba en un archivo de texto una línea con el texto que se indica a continuación para cada caso.

"El libro de código N está disponible para su préstamo"

"El libro de código N está prestado y lamentablemente ya fue reservado"

"El libro de código N está prestado pero no fue reservado"

"El libro de código N no existe en la biblioteca"

En el caso que un código N figure como "prestado" pero no está "reservado" entonces el programa deberá agregar ese código N al archivo de reservados.

Debe implementar toda primitiva que utilice. Indique ejemplos significativos para los casos de prueba.

Ejercicio 26: En este ejercicio se evaluará la correcta división del problema en subproblemas. Parcial 2014

Una empresa de turismo ofrece distintos destinos de viajes donde cada destino se encuentra codificado por un número entero. Se dispone de un **archivo de texto** viajes.dat donde cada línea contiene el siguiente formato de código del viaje seguido del cupo máximo de personas y la fecha en que se realiza el viaje en formato dd/mm/aaaa. Por ejemplo:

439	50	16/07/2014	<EOLN>
456	35	04/08/2014	<EOLN>
488	20	09/09/2014	
...			
<EOF>			

Donde el viaje 439 tiene un cupo para 50 personas y sale el 16 de julio de 2014 y el viaje 456 tiene cupo para 35 personas y sale el 4 de agosto de 2014, y así sucesivamente.

También se dispone de otro archivo de enteros reservas.dat que registra las personas que están anotadas para cada viaje con el siguiente formato código del viaje seguido del código de la persona. Por ejemplo:

439	1001	456	1012	456	1007	439	1044	439	1015	...
-----	------	-----	------	-----	------	-----	------	-----	------	-----

Escribir un programa en Pascal que leyendo la información de los archivos anteriormente mencionados, permita al operador del sistema realizar estas tareas:

- (1) dado un código de viaje mostrar su cupo máximo, la cantidad de lugares ocupados y de lugares disponibles.
- (2) mostrar en forma numerada los códigos y fecha de aquellos viajes que han completado su cupo.
- (3) Salir del sistema

```

Ingrese opción (1) viaje (2) completos (3) salir > 1
Ingrese el código del viaje: 456
Cupo: 35 Ocupados: 27 Disponibles: 7

Ingrese opción (1) viaje (2) completos (3) salir > 2
1. código 439 fecha viaje 16/07/2014
2. código 488 fecha viaje 09/09/2014

Ingrese opción (1) viaje (2) completos (3) salir > 3
Ejecución finalizada. Gracias.

```