



Trabajo Práctico 5

Funciones y Procedimientos.

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

Observación general 1: 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.

Observación general 2: 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. 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;
```

¿Es correcto suponer que la siguiente expresión debería ser siempre verdadera? $\text{Invertir}(\text{Invertir}(\text{Num})) = \text{Num}$

Considere: (A) $\text{invertir}(\text{invertir}(1234)) = 1234$ y (B) $\text{invertir}(\text{invertir}(1030)) = 1030$

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

- function** potencia(base, exponente: integer): integer; { Calcula $\text{Base}^{\text{Exponente}}$ }
- function** primo(nro: integer):boolean; { Determina si Nro es primo }.
- function** techo(nro: real): integer; { Calcula $\lceil \text{Nro} \rceil$ }
{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 `dia`, `mes` y `anio`. Por ejemplo si `dia= 21`, `mes= 10` y `anio=2008`, la fecha es válida. Si `dia=29`, `mes=2` y `anio = 2010` la fecha no es válida. Utilice para determinar la validez de la fecha la sentencia CASE.}



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. Una fecha es palíndromo si al concatenar los números que forman la fecha el número que se forma es capicúa. El número que se genere a partir de la fecha dependerá de cómo se manipulen sus campos de día, mes y año. Los formatos típicos son (1) día, mes y año; (2) mes, día y año; (3) año, mes y día. Por ejemplo, fecha 10/05/2015 bajo el formato (1) genera el número 1052015, bajo el formato (2) genera el número 5102015 y bajo (3) genera el número 2015510. Observe que el cero a la izquierda de los campos de fecha ingresada se pierde.

- Realice una primitiva que dado tres valores enteros positivos que representan una fecha valida (no es necesario verificarlo) y un entero entre 1 y 3 representando el formato típico. Retorne un número que resulte de concatenar los campos de la fecha según dicho formato.
- Realice una primitiva que determine si una fecha es palíndromo según alguno de los tres formatos. Deberá retornar 0 si ninguno de los formatos forma un palíndromo, y el código del formato (1, 2 o 3) si bajo alguno de los formatos forma palíndromo. Si fuese el caso que bajo más de un formato se forme palíndromo, puede retornar cualquiera de los códigos bajo los cuales se formen palíndromo. Esta primitiva deberá hacer uso de la primitiva anterior y de la función invertir.
- Realice un programa que permita seleccionar al usuario el formato deseado para ingresar una fecha y que luego el usuario ingrese la fecha como una secuencia de caracteres con separadores. Por ejemplo, si selecciona el formato 1 el usuario puede ingresar 21/03/1231. El programa deberá producir como resultado si la fecha ingresada es palíndromo bajo alguno de los formatos. Por ejemplo, dicha fecha es palíndromo bajo el formato (3) 1231321, pero no bajo el formato (1) 2131231 ni bajo el formato (2) 3211231.

Ejercicio 6. 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 7. 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.
- Considere el programa "parcial2". 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 parcial2;
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 8. 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.

Ejercicio 9. Archivos de Texto

- Escriba un procedimiento que indique cuantas líneas tiene un archivo de texto.
- Escriba una función que busque cuantas veces está un carácter E (ingresado por el usuario) en cada línea de un archivo de texto.
- Escriba un procedimiento que agregue dos líneas en blanco a un archivo de texto. ¿Podría utilizarse una función para dicha tarea? Justifique su respuesta.
- 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 10. Indique cuantos parámetros por valor y cuantos por referencia hay en cada uno de los siguientes procedimientos y funciones:

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



Ejercicio 11. Conceptos

- 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 12.

- (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 13. 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 14. (Parcial 2009) 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 15. 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 16. Considere la siguiente definición de la función invertir. ¿Qué problemas tiene?

```
function invertir( num: integer ):integer;
begin
  invertir := 0;
  while num > 0 do
  begin
    invertir := invertir * 10 + num mod 10;
    num := num div 10;
  end;
end;
```

Ejercicio 17. 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 18. En este ejercicio se evaluará la correcta **división del problema en subproblemas**. (Recuperatorio 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.
```

Ejercicio 19. 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 20. En este ejercicio se evaluará la correcta **división del problema en subproblemas**. (Parcial 2015)

Considere una empresa que vende productos tanto en sus sucursales como por una aplicación web. Cada vez que se vende un producto en una sucursal, el código del producto (un número entero positivo) se almacena en el archivo `SUC.ventas` y cada vez que un producto se vende por internet, su código se almacena en otro archivo `WEB.ventas`. Además la empresa tiene un tercer archivo de enteros positivos, llamado `productos.cod`, con todos los códigos de productos que vende (en este archivo no hay elementos repetidos).

Se debe escribir un programa en Pascal que, utilizando de los archivos indicados antes, permita realizar estas tareas:

- (1) Ingresar un código de producto y mostrar por pantalla por cuál medio se vendió más, si el producto no se vendió por ningún medio o se vendió igual cantidad debe indicarse.



(2) Utilizando los códigos del archivo `productos.cod` generar un reporte por pantalla de los productos que no se vendieron por ningún medio.

Ejemplo `Productos.cod`: 100, 101, 102, 103, 104, 105, 106
`SUC.ventas`: 103, 100, 101, 104, 101, 104, 105, 103, 104
`WEB.ventas`: 103, 104, 100, 105, 103, 105, 100

```
(1) Mejor Medio (2) Productos sin venta (3) Salir > 1
Ingrese código de producto: 100
El producto se vendió más por WEB.
(1) Mejor Medio (2) Productos sin venta (3) Salir > 1
Ingrese código de producto: 101
El producto se vendió más por SUCURSALES.
(1) Mejor Medio (2) Productos sin venta (3) Salir > 1
Ingrese código de producto: 102
El producto no se ha vendido.
```

```
(1) Mejor Medio (2) Productos sin venta (3) Salir > 1
Ingrese código de producto: 103
El producto se vendió por igual en la WEB como en SUC.
(1) Mejor Medio (2) Productos sin venta (3) Salir > 2
Productos que no fueron vendidos: 102 106
(1) Mejor Medio (2) Productos sin venta (3) Salir > 3
Ejecución Finalizada. Gracias.
```

Ejercicio 21. 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:

- (4) dado un código de viaje mostrar su cupo máximo, la cantidad de lugares ocupados y de lugares disponibles.
- (5) mostrar en forma numerada los códigos y fecha de aquellos viajes que han completado su cupo.
- (6) 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
3. código 439 fecha viaje 16/07/2014
4. código 488 fecha viaje 09/09/2014

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



Ejercicio 22. En este ejercicio se evaluará la correcta **división del problema en subproblemas**. (Recuperatorio 2015)

En un reactor debe mantenerse la temperatura de dos tanques de enfriamiento (llamados Tanque1 y Tanque2) entre los valores "V1" y "V2" ($V1 < V2$). Cada tanque tiene dos sensores que toman la temperatura cada 1 segundo y la almacenan en un archivo. Por lo tanto el sistema tiene cuatro archivos de números **reales** "sensor1deTanque1", "sensor2deTanque1", "sensor1deTanque2", y "sensor2deTanque2". Un valor de temperatura en un archivo estará fuera del rango permitido, si el valor es menor a V1 o mayor a V2.

Se debe escribir un programa en Pascal que, utilizando los archivos indicados antes, permita realizar estas tareas:

- (1) Ingresar un número de tanque y el programa deberá dar una alerta ROJA si en los dos archivos de sensado de temperatura para ese tanque hay más de 5 valores fuera de rango, una alerta NARANJA si en uno de los sensores la cantidad de valores fuera de rango es mayor a 5 y en el otro sensor la cantidad es entre 2 y 5, una alerta AMARILLA si en al menos uno de los dos archivos hay entre 2 y 5 valores fuera de rango y en el otro no hay una cantidad mayor a 5. De lo contrario deberá indicar "por el momento sin alertas".
- (2) Ingresar un tanque e indicar: cuál sensor mide más valores fuera de rango, si midieron igual cantidad de fuera de rango, o si ninguno midió fuera de rango (ningún valor fuera de rango).

En ambos casos verifique que el tanque ingresado es válido y respete el formato de los mensajes por pantalla.

EJEMPLO 1

V1 = 0 y V2= 10

T1S1: 1.5 2.3 4.5 7.8 9.9 9.7 9.4

T1S2: 1.6 2.2 4.2 7.9 9.9 9.8 9.5

T2S1: 2.4 8.3 12 25.2 45.3 32 24.2 16.8 11.9

T2S2: 2.5 11.1 15.1 27.3 47.8 35.6 23.9 17 12.6

EJEMPLO 2

V1 = 0 y V2= 10

T1S1: 6.7 10.5 7.8 8.9 12.1 10.2

T1S2: 6.2 9.8 7.1 10.2 11.5 9.4

T2S1: 45.1 48.3 51.2 60 65.8 78.4

T2S2: 5.2 6.8 9.3 55 65.9 79.1

(1) Alertas (2) Sensores (3) Salir > 1

Ingrese un número de tanque: 1

Por el momento sin alertas.

(1) Alertas (2) Sensores (3) Salir > 1

Ingrese un número de tanque: 2

Tanque 2 en alerta ROJA.

(1) Alertas (2) Sensores (3) Salir > 2

Ingrese un número de tanque: 1

Ninguno de los valores está fuera de rango.

(1) Alertas (2) Sensores (3) Salir > 2

Ingrese un número de tanque: 2

El sensor 2 midió 1 valor/es más que el sensor 1.

(1) Alertas (2) Sensores (3) Salir > 3

Ejecución Finalizada. Gracias.

(1) Alertas (2) Sensores (3) Salir > 1

Ingrese un número de tanque: 1

Tanque 1 en alerta AMARILLA.

(1) Alertas (2) Sensores (3) Salir > 1

Ingrese un número de tanque: 2

Tanque 2 en alerta NARANJA.

(1) Alertas (2) Sensores (3) Salir > 2

Ingrese un número de tanque: 1

El sensor 1 midió 1 valor/es más que el sensor 2.

(1) Alertas (2) Sensores (3) Salir > 2

Ingrese un número de tanque: 2

El sensor 1 midió 3 valor/es más que el sensor 2.

(1) Alertas (2) Sensores (3) Salir > 3

Ejecución Finalizada. Gracias.



Ejercicio 23. En este ejercicio se evaluará la correcta **división del problema en subproblemas**. (Recuperatorio 2013)

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 fue no 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.