



## Trabajo Práctico Nº 4

### I. Repetición incondicional

**Ejercicio 1.** Se desea realizar una aplicación que solicite al usuario un carácter y un número natural N, y que la aplicación muestre en pantalla dicho carácter repetido N veces consecutivas. *Ejemplo:*

```
Ingrese un carácter: *
Ingrese un número de repeticiones: 40
*****
```

**Ejercicio 2.** Se desea realizar una aplicación que solicite al usuario tres números naturales (que llamaremos A, B, y N), que muestre por pantalla todos los múltiplos de N que estén entre A y B inclusive. Asuma que los datos son ingresados correctamente, es decir, el usuario ingresa  $N \geq 0$ , y  $0 \leq A < B$ . Realice un algoritmo antes de escribir el programa, luego implemente en Pascal utilizando un ciclo FOR. ¿Qué casos de prueba usaría?

Ejemplo:

```
Ingrese un natural N: 4
Ingrese dos naturales para el rango [A..B]: 3 17
Los múltiplos de 4 entre 3 y 17 son: 4 8 12 16
```

**Ejercicio 3.** Se desea realizar una aplicación que solicite al usuario un número natural y que como resultado muestre por pantalla todos los divisores naturales de dicho número y cuantos divisores obtuvo. Realice un algoritmo antes de escribir el programa, luego implemente en Pascal utilizando un ciclo FOR. ¿Qué casos de prueba usaría?

Ejemplo:

```
Ingrese un entero: 116
Los divisores de 116 son: 1 2 4 29 58 116
La cantidad de divisores de 116 es: 7
```

**Ejercicio 4.** Para los siguientes programas realice una traza y determine que se mostrará en pantalla durante la ejecución de los mismos.

<pre>program ej4a; var a,i:integer; begin   a:= 10;   for i:= 0 to 0 do     a:= a + 1;   writeln('a= ', a) end.</pre>	<pre>program ej4b; var a,i:integer; begin   a:= 10;   for i:= 1 to -1 do     a:= a + 1;   writeln('a= ', a) end.</pre>	<pre>program ej4c; var a,i:integer; begin   a:= 10;   for i:= 1 to 0 do     a:= a + 1;   writeln('a= ', a) end.</pre>
<pre>program ej4d; var i:integer; begin   for i:= 5 to 3 do     writeln('i=', i) end.</pre>	<pre>program ej4e; var a,i:integer; begin   a:= 1;   for i:= 1001 to 1200 do     a:= a + 1;   writeln('a= ', a) end.</pre>	<pre>program ej4f; var i:integer; begin   for i:= 5 downto 3 do     writeln('i=', i) end.</pre>



**Ejercicio 5.** Considerando el siguiente programa

```

program ej25b;
var v,a,b:integer;
begin
    a := 1;
    b := 4;
    for v := a+3 to a*b*b do
        writeln('v = ', v);
end.
    
```

- i. ¿Cuántas veces se ejecuta la sentencia `writeln('v = ', v)` ?
- ii. ¿Por qué no es necesario poner el `writeln` dentro de una sentencia compuesta `begin ... end`?
- iii. ¿Es posible modificar el valor de las variables `v`, `a` o `b` dentro del FOR?
- iv. ¿En qué momento se conoce cuántas veces se va a repetir la sentencia dentro de un ciclo FOR? (1) en compilación, (2) antes de comenzar el `for`, (3) se evalúa en cada iteración. Fundamente su respuesta.

**Ejercicio 6.** Durante la ejecución del siguiente programa

```

Program ej6;
var i,j: integer;
begin
    for i := 1 to 10 do
        begin
            for j := 1 to i do
                write ( i * j );
                writeln
            end;
        end.
    end.
    
```

- a) ¿Cuántas veces se ejecuta la instrucción `write`?
- b) ¿Cuántas veces se ejecuta la instrucción `writeln`?
- c) ¿Qué se muestra en pantalla?

**Ejercicio 7.** Escriba un programa para calcular las siguientes sumatorias:

a) $\sum_{i=0}^n 1$	c) $\sum_{i=0}^n \sum_{j=1}^m 1$	e) $\sum_{i=0}^n \sum_{j=1}^m (i + j)$
b) $\sum_{i=0}^n i$	d) $\sum_{i=0}^n \sum_{j=1}^m i$	f) $\sum_{i=0}^n \sum_{j=i}^m (i + j)$

Observación: si no conoce esta notación puede ver una explicación introductoria en <http://es.wikipedia.org/wiki/Sumatorio>

**Ejercicio 8.** Escriba un programa que dados como entrada un valor N muestre N filas de la forma 1 2 ... N, 2 3 ... N+1, ... etc. Por ejemplo, si se ingresa el valor 5 se debería mostrar:

```

1 2 3 4 5
2 3 4 5 6
3 4 5 6 7
4 5 6 7 8
5 6 7 8 9
    
```



**Ejercicio 9.** Escriba un algoritmo para calcular la suma de los divisores positivos de un número **N**. Por ejemplo si  $N=12$ , sus divisores son 1, 2, 3, 4, 6 y 12, y la suma de los divisores es 28. El número  $N=17$  es primo, por lo tanto sus únicos divisores son 1 y 17, y la suma es 18. ¿Qué otros casos de prueba usaría?

Ejemplo:

Ingrese un entero: 12

La suma de los divisores de 12 es: 28

**Ejercicio 10. Conceptos Teóricos**

- a) ¿Qué es un diagrama sintáctico? ¿Cuál es su utilidad para los lenguajes de programación? Muestre como ejemplo el diagrama sintáctico de la sentencia FOR.
- b) Explique los términos: hardware y software.
- c) Indique cuatro pautas de buena programación que ayudan al desarrollo, mantenimiento y futuras actualizaciones del software.
- d) Considere la sentencia `FOR v:= inicio TO fin DO write(v);` ¿Cuántas veces se repetirá `write(v)`?

**Ejercicio 11.** Suponga que  $1 \leq a \leq 9$ ,  $0 \leq b \leq 9$ , y los números  $n$  y  $m$  son naturales.

- a) Escriba un programa para mostrar por pantalla todos los números de la forma *abab*. Los números deberán mostrarse por renglón y numerados.
- b) Escriba un programa para leer por teclado dos números  $n$  y  $m$  y mostrar por pantalla todos los números de la forma *abab*, con  $1 \leq a \leq 9$ ,  $0 \leq b \leq 9$ , que sean divisibles por  $n$  y  $m$ . Los números deberán mostrarse por renglón y numerados. Ej: si  $n=2$  y  $m=5$  debería verse en pantalla lo siguiente:

- |         |   |
|---------|---|
| 1. 1010 | Proponga casos de prueba adicionales para el inciso (b) donde: 1) Exista algún                  |
| 2. 2020 | número de la forma <i>abab</i> que sea divisible por $n$ pero no divisible por $m$ . (2) Exista |
| 3. 3030 | algún número de la forma <i>abab</i> que sea divisible por $m$ pero no divisible por $n$ . (3)  |
| .....   |   |
| 9. 9090 | Ningún número de la forma <i>abab</i> sea divisible por $n$ y por $m$ . (4) Ningún número de    |
|         | la forma <i>abab</i> sea divisible por $n$ o por $m$ .  |

**Ejercicio 12. (Buffer - FOR):** Escriba un programa en Pascal para resolver cada uno de los siguientes ejercicios:

- a) Leer una secuencia de números 5 naturales, y mostrar el menor y el mayor de todos los números leídos. Pruebe su programa con (a) todos los nros diferentes, (b) todos nros iguales, (c) algunos nros iguales y que se corresponden con el menor de la secuencia.
- b) Leer una secuencia de números determinada por una cantidad especificada por el usuario y mostrar todos aquellos que sean negativos. Pruebe con una secuencia donde (a) haya algunos negativos, (b) todos sean negativos, (c) ninguno sea negativo.
- c) Leer una secuencia de caracteres determinada por una cantidad especificada por el usuario y mostrar la cantidad de vocales. ¿Cuáles serían los casos de prueba a utilizar?
- d) Leer una secuencia de caracteres determinada por una cantidad especificada por el usuario y mostrar cada carácter leído en un línea diferente seguido de 5 caracteres ascii posteriores consecutivos. *Por ejemplo,*

*Ingrese la cantidad de caracteres a leer: 4*

*Ingrese la secuencia: adtb<ENTER>*

*abcde*

*defgh*

*tuvwx*

*bcdef*



## II. Repetición condicional.

**Ejercicio 13.** Modifique el programa realizado en el práctico 3 ejercicio 4 inciso f realizando una validación de datos (mientras no sea correcto seguir pidiendo los datos).

**Ejercicio 14.** Para cada uno de los siguientes enunciados, seleccione adecuadamente una sentencia de repetición y escriba un programa que permita:

- Determinar si un dígito D aparece en un número entero N. Ejemplos: si D = 3 y N=1234, entonces D aparece; si D = 6 y N=661, entonces D aparece; si D = 3 y N=661, entonces D NO aparece.
- Para un número de más de 5 cifras (*i.e.*, mayor a 9999), retornar la suma de las últimas 5 cifras. Realice una validación sobre la entrada.
- Para un número cualquiera, retornar la suma de las últimas N cifras (si existen), donde N es ingresado por el usuario. Si no existen una cantidad suficiente de cifras deberá mostrar un mensaje indicando el hecho y solicitando un nuevo número. Determine casos de prueba.

*Observación: la abreviatura "i.e." muy usada en textos científicos, proviene del latín "id est" que en castellano sería: "es decir" o "esto es". ([es.wikipedia.org/wiki/Anexo:Abreviaturas\\_latinas\\_en\\_bibliografia\\_cientifica](http://es.wikipedia.org/wiki/Anexo:Abreviaturas_latinas_en_bibliografia_cientifica))*

**Ejercicio 15. (Buffer):** Escriba un programa en Pascal para resolver cada uno de los siguientes ejercicios:

- Leer una secuencia de números naturales terminada en 0 y mostrar el promedio. Pruebe su programa con una secuencia que consiste sólo del 0 (en cuyo caso no hay promedio), y luego con un par de secuencias que tengan una cantidad de elementos mayor a cero.
- Leer una secuencia de números naturales terminada en 0 y mostrar una secuencia con los valores leídos que tienen más de tres cifras. Además de la secuencia vacía, pruebe su programa con una secuencia que no tenga números de tres cifras, y con alguna otra secuencia que considere útil.
- Leer una secuencia de caracteres terminada en un punto y contar cuántos de ellos son espacios en blanco (carácter ' '). *"Por ejemplo, en esta misma oración que está leyendo hay 13 espacios en blanco"*. Defina los casos de prueba útiles.
- Leer una secuencia de caracteres terminada en un punto, y contar la cantidad de letras (A hasta la Z incluyendo la Ñ y las vocales con acentos, tanto mayúsculas como minúsculas). Por ejemplo, en la oración siguiente hay 12 letras. *Día 23 (cena): ñoquis*. Defina casos de prueba, donde al menos unos de los casos produzca un resultado de 0 letras.

**Ejercicio 16.** Indicar el valor final de cada variable presente en los siguientes programas.

(1)	(2)	(3)	(4)	(5)
<pre> program ej4a; var a,b:integer; begin   a := 0;   b := 1;   while b &lt; 90 do     b := b * 2; end. </pre>	<pre> program ej4b; var c:integer; begin   c := 100;   repeat     c := c - 1   until c = 0; end. </pre>	<pre> program ej4c; var a:integer; begin   a := 9;   while a &lt; 10 do     a := a + 1; end. </pre>	<pre> program ej4d; var a:integer; begin   a := -1;   repeat     a := a + 1   until a = 0; end. </pre>	<pre> program ej4e; var b:integer; begin   b := 5;   while (b-1)=0 do   b := b - 1; end. </pre>



**Ejercicio 17.** Para cada uno de los siguientes enunciados, seleccione adecuadamente una sentencia de repetición y escriba un programa que permita:

- Sumar los primeros  $N$  números naturales, para un  $N$  ingresado por el usuario.
- Calcular el producto de los primeros  $N$  naturales pares, para un  $N$  ingresado por el usuario.
- Indicar si un entero positivo  $N$  es o no un número **primo**.

Para cada inciso indique los casos de prueba a utilizar.

*Definiciones: Un **número primo** es un número natural que tiene únicamente dos divisores naturales distintos: él mismo y el 1. Todo número natural no primo, a excepción del 1, se denomina compuesto, es decir, tiene uno o más divisores distintos a 1 y a sí mismo.. El número 1, no se considera ni primo ni compuesto.*

[http://es.wikipedia.org/wiki/Número\\_primo](http://es.wikipedia.org/wiki/Número_primo)

**Ejercicio 18. (conceptos)**

- ¿Por qué decimos que REPEAT y WHILE son repeticiones condicionales, pero en cambio FOR es una repetición incondicional?
- ¿Qué diferencias hay entre las repeticiones con WHILE, REPEAT y FOR? Acompañe su respuesta con ejemplos simples que muestren lo que está indicando.
- ¿Qué similitudes encuentra entre las sentencias repetitivas WHILE, FOR y REPEAT?

d) Considere el programa “ejemplo” de la derecha. Escriba otro código fuente que tenga el mismo efecto al dado (para cualquier valor de la variable entrada) pero reemplazando la sentencia while por otra sentencia repetitiva que no sea while.

```

program ejemplo;
var entrada: integer;
begin
    write('ingrese entero');
    readln(entrada);
    while (entrada <> 0) do
    begin
        writeln(entrada);
        write('ingrese entero');
        readln(entrada);
    end; // del while
end.
    
```

- ¿Con qué sentencias se puede dar un ciclo infinito? De dos ejemplos en Pascal.
- En un programa en Pascal, ¿puedo siempre reemplazar una sentencia FOR por una sentencia WHILE y obtener un código ejecutable que tenga el mismo efecto? ¿Qué tengo que tener en cuenta? Muestre un ejemplo concreto en Pascal.
- En un programa en Pascal, ¿Siempre es posible reemplazar un WHILE por un FOR y obtener un código ejecutable que tenga el mismo efecto? Justifique si es que se puede, o muestre un ejemplo concreto en Pascal donde se vea que no es posible.
- En un programa en Pascal, ¿puedo siempre reemplazar una sentencia WHILE por una sentencia REPEAT y obtener un código ejecutable que tenga el mismo efecto? ¿Qué modificaciones son necesarias para que el programa modificado tenga el mismo efecto que el original? ¿Qué tengo que tener en cuenta? Muestre un ejemplo.



- i) ¿Y reemplazar un FOR por un REPEAT y obtener un código ejecutable que tenga el mismo efecto? Muestre un ejemplo.

### **III. Entrada/Salida con Archivos de Datos.**



**Ejercicio 19.** Escriba un programa que muestre en pantalla el contenido de un archivo de enteros. Suponga que dicho archivo ya se encuentra creado, a tal efecto puede utilizar el generador de archivos provisto por la cátedra. Proponga casos de prueba.



**Ejercicio 20.** Escriba un programa que indique si el archivo de números enteros llamado "enteros.dat" tiene elementos o está vacío.



**Ejercicio 21.** Dado un archivo "enteros.dat" ya creado y con números enteros en él, escriba un programa que cuente la cantidad de elementos que tiene dicho archivo.



**Ejercicio 22.** Escriba un programa que busque cuantas veces está el entero  $E$  (ingresado por el usuario) en el archivo "enteros.dat" (ya creado y con números en él). Proponga casos de prueba.

#### **Ejercicio 23. (Conceptos)**

a) Indique al menos 4 características importantes del tipo de dato FILE.

b) Indique 3 características que poseen las variables de tipo simple en Pascal (integer, real, char, boolean) con respecto a su almacenamiento en memoria. Luego indique que ocurre con el almacenamiento en memoria de los archivos secuenciales (file).

**Ejercicio 24. (Buffer):** Escriba un programa en Pascal para resolver cada uno de los siguientes ejercicios:

a) Leer una secuencia de números enteros terminada en 0 y mostrar el mayor.

b) Leer una secuencia de pares números enteros terminada en 0 0 y mostrar una secuencia formada por los mismos pares de valores pero ordenada de manera que siempre aparezca primero el mayor y luego el menor elemento del par.

c) Leer una secuencia de caracteres terminando con un punto y mostrar la cantidad de vocales mayúsculas.



**La siguiente definición es necesaria para poder resolver correctamente varios incisos a continuación:**

**Definición:** En un sistema de numeración posicional el número de símbolos permitidos se conoce como base del sistema. Por ejemplo, en el sistema decimal los símbolos válidos son 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 por lo cual la base es 10, en un sistema binario los símbolos válidos son 0 y 1 por lo tanto su base es 2, el sistema octal tiene base 8, el duodecimal tiene base 12, el hexadecimal tiene base 16, el sexagesimal tiene base 60, etc. El teorema fundamental de la numeración establece que un número  $N$  con cantidad finita de decimales respeta las siguientes igualdades:

$$N = d_n \dots d_1 d_0, d_{-1} \dots d_{-k} =$$

$$d_n \cdot 10^n + \dots + d_1 \cdot 10^1 + d_0 \cdot 10^0 + d_{-1} \cdot 10^{-1} + \dots + d_{-k} \cdot 10^{-k} =$$

$$N = \sum_{i=-k}^n d_i \cdot 10^i$$

Fuente: [http://es.wikipedia.org/wiki/Sistema\\_de\\_numeración](http://es.wikipedia.org/wiki/Sistema_de_numeración)

Utilizaremos la definición anterior como base para armar números enteros o decimales a partir de sus dígitos o para desarmarlo según sea necesario. Por lo cual, diremos que la unidad se encuentra en la posición 0, la decena en la posición 1, la centena en la posición 2, y así sucesivamente. Podemos definir entonces **que la posición de cualquier dígito de un número  $N$  será igual a la potencia que se utilizará en la base al multiplicar por dicho dígito.** Por ejemplo, si  $N=1234 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$ , por lo tanto la posición del dígito 4 es 0 y la del dígito 3 es 1, la del 2 es 2 y la posición del dígito 1 es 3.

Salvo que el enunciado indique expresamente otra cosa, en lo que sigue se utilizará la anterior definición de posición de un dígito en un número.

**Ejercicio 25. (Descomposición numérica):** Para cada uno de los siguientes enunciados escriba un programa que dado un número entero  $N$  provisto por el usuario (deberá ser solicitado en el mismo programa) permita:

- Sumar todos sus dígitos. Ej: si ingresa 343 deberá mostrar 10; si ingresa -20 mostrará 2. Pruebe su programa con un número de un único dígito y luego con algún número de más dígitos.
- Sumar todos los dígitos que se encuentran en **posiciones pares (ver definición anterior)**. Ej. si ingresa 343 deberá mostrar 6; si ingresa -27 mostrará 7.
- Encontrar el mayor dígito presente en  $N$ . Pruebe su programa con un número con todos sus dígitos diferentes y con otro número donde el mayor se encuentre varias veces.
- Determinar si un dígito  $D$  está presente en  $N$ . Ej: el dígito 5 está presente en el entero 345; el dígito 3 no está presente en el entero -122. Defina casos de prueba.
- Contar cuantos dígitos impares tiene  $N$ . Defina casos de prueba.
- Contar cuantos dígitos impares se encuentran en **posiciones pares**.



**Ejercicio 26.** Para los siguientes incisos realice una versión equivalente a la ya realizada utilizando una estructura repetitiva diferente a la que usó entonces.

- Sumar los primeros  $N$  números naturales, para un  $N$  ingresado por el usuario.
- Determinar si un dígito  $D$  está presente en  $N$ . Ej: el dígito 5 está presente en el entero 345; el dígito 3 no está presente en el entero -122.

**Ejercicio 27.** Escriba un programa que lea por teclado un número natural  $N$  y determine si es o no capicúa. Un número  $N$  formado por dígitos " $d_k d_{k-1} \dots d_1 d_0$ " es capicúa si es igual al número " $d_0 d_1 d_2 \dots d_k$ ".

**Ejercicio 28.** Escriba un programa que lea por teclado un número natural  $N$  y determine si la suma de los dígitos de  $N$  en **posiciones impares** es igual a la suma de los dígitos de  $N$  en **posiciones pares**. Asuma que si  $N$  consiste de un solo dígito la suma de los dígitos de posiciones impares es cero. Defina casos de prueba.

 **Ejercicio 29.** Escriba un programa que dados dos archivos de números enteros ya creados "enteros1.dat" (F1) y "enteros2.dat" (F2), junte el contenido de ambos archivos en un tercer archivo nuevo "enterosly2.dat" (F3), de forma tal que en F3 queden intercalados un elemento de F1 y F2 mientras sea posible.

 **Ejercicio 30.** Un archivo "enteros.dat" ya creado y con números enteros en él contiene las notas de un curso. Se conoce cuantos alumnos tiene el curso, pero se desconoce la cantidad de notas almacenadas en el archivo. También se sabe que todos los alumnos recibieron la misma cantidad de notas y que todas las notas de un mismo alumno están ubicadas consecutivamente en el archivo. Escriba un programa para calcular el promedio de las notas de cada alumno. Deberá solicitar al usuario solamente la cantidad de alumnos del curso (y supondremos que la cantidad de elementos del archivo es múltiplo de dicha cantidad).

**Ejercicio 31.** Para saber si un número natural  $n$  es divisible por 11, basta hacer lo siguiente:

- Calcular la suma de todos los dígitos que ocupan posiciones impares en el número  $n$ .
- Calcular la suma de todos los dígitos que ocupan posiciones pares en el número  $n$ .
- Hallar la diferencia entre las sumas anteriores.
- Si el valor absoluto de dicha diferencia es 0, 11, 22 ó 33, el número  $n$  es divisible por 11.

Escriba un programa para determinar si un número entero es divisible por 11, aplicando el criterio descripto.

**Ejercicio 32.** Considerando el siguiente programa:

```

program Sorpresa;
var dato, aux, dig: INTEGER;
begin
    read(dato);
    aux := dato;
    repeat
        dig := dato mod 10;
        dato := dato div 10;
        aux:= aux * 10 + dig
    until dato = 0;
    write(aux)
end.

```

- ¿Qué operación realiza el programa Sorpresa?. Confeccione trazas con  $aux = 4834$  y  $aux = 4830$
- ¿Qué ocurre si se cambia  $aux:=dato$  por  $aux:=0$ ?



**Ejercicio 33.** Suponga que  $S_i$  se refiere a una sentencia válida en Pascal (sintáctica y semánticamente). ¿Qué error tienen las siguientes sentencias?

<b>(i)</b>	<b>(ii)</b>	<b>(iii)</b>
<pre>while X &gt; Y do   S1; S2; ... end;</pre>	<pre>while no eof do S1;</pre>	<pre>for j := M down N do S1;</pre>
<b>(iv)</b>	<b>(v)</b>	<b>(vi)</b>
<pre>for A := 2.5 to 13.0 do S2;</pre>	<pre>for J := 1 to 10 do   for J := 5 downto 1 do     write(J);</pre>	<pre>for J := 1 to 10 do   if J = 5 then     J := 10;</pre>

**Ejercicio 34. (este ejercicio fue extraído de un parcial):**

Considere que dispone de un archivo de números enteros llamado **mis\_enteros.dat**. Escriba un programa en Pascal que solicite al usuario dos números enteros *Limite1* y *Limite2*, y que luego elimine del archivo todos los elementos cuyo valor esté entre *Limite1* y *Limite2*. Además se deberá mostrar por pantalla cuántos elementos fueron eliminados.

Ejemplo: Si el archivo **mis\_enteros.dat** tiene los números: 41 34 2 -23 4 35 93 -4 20

Si el usuario ingresa *Limite1* = 40 y *Limite2* = 20, entonces deberá eliminar 34, 35 y 20 y el programa mostrará "Se eliminaron 3 elementos"

Si el usuario ingresa *Limite1* = -10 y *Limite2* = 5, entonces deberá eliminar 2, 4 y -4 y el programa mostrará "Se eliminaron 3 elementos"

Si el usuario ingresa *Limite1* = 100 y *Limite2* = -40, entonces elimina todos y el programa mostrará "Se eliminaron 9 elementos"

Si el usuario ingresa *Limite1* = 10 y *Limite2* = 15, no eliminará ninguno y el programa mostrará "Se eliminaron 0 elementos"



## **IV. Archivos de Texto**

**Ejercicio 35.** Compare los tipos de datos FILE OF CHAR y TEXT.

- ¿Qué similitudes existen entre dichos tipos de datos?
- ¿Qué diferencias existen y es necesario tener en cuenta al momento de escribir un programa?

**Ejercicio 36.** Escriba un programa que cuente cuantas veces está un carácter ingresado por el usuario en un archivo de texto.

**Ejercicio 37.** Escriba un programa que cuente cuantas oraciones hay en un archivo de texto.

**Ejercicio 38.** Escriba un programa que indique cuantas líneas tiene un archivo de texto.

**Ejercicio 39.** Escriba un programa que busque cuantas veces está un carácter ingresado por el usuario en cada línea de un archivo de texto.

**Ejercicio 40.** Escriba un programa que modifique un archivo de texto de forma que cada oración comience en una nueva línea.

**Ejercicio 41.** Escriba un programa que reemplace un carácter ingresado por el usuario por otro carácter también ingresado por el usuario en un archivo de texto, indique en cuantas líneas lo reemplazó.

**Ejercicio 42.** Escriba un programa que busque cuantas veces está el carácter 'A' seguido del carácter 'C' en un archivo de texto.

**Ejercicio 43.** Escriba un programa que lea un archivo de texto con pares LU Nota (donde Nota es A, B, C, D, o E) 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, B o C, y Desaprobado si es D o E.

**Ejercicio 44.** 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).