



Trabajo Práctico Nº 5 Repetición y Archivos

Ejercicio 1: Indicar el valor final de cada variable presente en los siguientes fragmentos de programa.

(1)	(2)	(3)	(4)	(5)
<pre>a := 0; b := 1; while b < 90 do b := b*2;</pre>	<pre>c := 100; Repeat c := c - 1 until c = 0;</pre>	<pre>a := 9; while a < 10 do a := a + 1;</pre>	<pre>a := -1; Repeat a := a + 1 until a = 0;</pre>	<pre>b := 5; while b = 0 do b := b - 1;</pre>

Ejercicio 2: 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.

*Def: Un **número primo** es un número natural que tiene únicamente dos divisores naturales distintos: él mismo y el 1. Los números compuestos, que son aquellos que tienen algún divisor natural aparte de él mismo y del 1. El número 1, por convenio, no se considera ni primo ni compuesto. Fuente:*

http://es.wikipedia.org/wiki/Número_primo

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

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 3: 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.
- Sumar todos los dígitos que se encuentran en **posiciones pares (ver definición anterior)**. Ej. si ingresa 343 deberá mostrar 6; si ingresa -20 mostrará 0.
- Encontrar el mayor dígito presente en N .
- 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.
- Contar cuantos dígitos impares tiene N .
- Contar cuantos dígitos impares se encuentran en **posiciones pares**.

Ejercicio 4: Para los incisos **2.a.**, **2.c** y **3.d** realice una versión equivalente a la ya realizada utilizando una estructura repetitiva diferente a la que usó entonces.

Ejercicio 5:

- ¿Por qué decimos que REPEAT y WHILE son repeticiones condicionales?
- ¿Qué diferencias hay entre las repeticiones con WHILE, REPEAT y FOR? Acompañe su respuesta con ejemplos simples que muestren lo que está indicando.
- En un programa, ¿puedo reemplazar un WHILE por un REPEAT? ¿Qué modificaciones son necesarias para que el programa modificado tenga el mismo efecto que el original? ¿Qué tengo que tener en cuenta?



Ejercicio 6: 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.*



Ejercicio 7: Escriba un programa que indique si el archivo de números enteros "enteros.dat" tiene elementos o está vacío.



Ejercicio 8: 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 9: 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).



Ejercicio 10: Considere el siguiente programa que cuenta la cantidad de dígitos pares e impares de un número ingresado por el usuario. *Obs: Existe alguna controversia sobre si el número cero es considerado como natural, sin embargo, es considerado como entero y por lo tanto es considerado como par.*

```
Program parimpar;
var n, digito, aux: integer;
    pares, impares: integer;
begin
    pares := 0;
    impares := 0;
    readln(n);
    aux := n;
    repeat
        digito := aux mod 10;
        if (digito mod 2 = 0) then
            pares := pares + 1
        else
            impares := impares + 1;
        aux := aux div 10;
    until aux = 0;
    writeln('Existen ',pares,' dígitos pares en ',n);
    writeln('Existen ',impares,' dígitos impares en ',n);
end.
```

- Obtenga una versión equivalente modificando el programa para utilizar una estructura repetitiva **while do** en lugar de la estructura **repeat until**.
- ¿Se podría evitar el uso de la variable `aux` y en su lugar utilizar directamente la variable `n`?
- Verifique qué sucede en ambas versiones cuando el usuario ingresa el número cero.

Ejercicio 11: 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 12:** 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 13:** 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 14: 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.

Ejercicio 15: Escriba un programa que realice la **división entera** de dos números enteros positivos mediante restas sucesivas. El programa deberá mostrar por pantalla el resultado y el resto de la división.



Ejercicio 16: 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 17: El criterio de divisibilidad del 11 dice que, 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 descrito anteriormente.

Ejercicio 18: Dados los siguientes bloques de instrucciones, realizar una traza para determinar qué valor se imprimirá por pantalla al finalizar la ejecución de los mismos.

- | | | |
|---|--|---|
| a) <code>a:= 10;</code>
<code>for i:= 0 to 0 do</code>
<code> a:= a + 1;</code>
<code>writeln('a = ', a)</code> | b) <code>a:= 10;</code>
<code>for i:= 1 to -1 do</code>
<code> a:= a + 1;</code>
<code>writeln('a = ', a)</code> | c) <code>a:= 10;</code>
<code>for i:= 1 downto 0 do</code>
<code> a:= a + 1;</code>
<code>writeln('a = ', a)</code> |
| d) <code>for i:= 5 to 3 do</code>
<code> writeln(i);</code> | e) <code>for i:= 5 downto 3 do</code>
<code> writeln(i);</code> | |

Ejercicio 19:

- ¿Es posible en un programa reemplazar un WHILE por un FOR? Justifique.
- Considerando la declaración "VAR v, a, b: integer;" y que las variables **a** y **b** tienen los valores 1 y 4 respectivamente. En la sentencia:


```

for v:= a+3 to a*b*b do
begin
    write(v); writeln;
end;

```

 - ¿Cuántas veces se ejecuta `write(v)`?
 - ¿Es posible modificar el valor de las variables **v**, **a** o **b** dentro del FOR?
 - ¿En que momento de la ejecución se conoce cuántas veces se va a repetir el bloque? (1) en compilación, (2) antes de comenzar el `for`, (3) nunca. Fundamente su respuesta.



- c) Considere la sentencia `FOR v:= inicio TO fin DO write(v);` ¿Puede escribir una expresión que indique cuántas veces se repetirá `write(v);`?

Ejercicio 20: Durante la ejecución del siguiente bloque de instrucciones:

```
for I := 1 to 10 do
begin
  for J := 1 to I do
    write ( I * J );
  writeln
end;
```

- ¿Cuántas veces se ejecuta la instrucción `write`?
- ¿Cuántas veces se ejecuta la instrucción `writeln`?
- ¿Cuál es el último valor mostrado?

Ejercicio 21: El criterio de divisibilidad del 9 dice que, para saber si un número natural n es divisible por 9, basta hacer lo siguiente:

- Calcular la suma de todos los dígitos del número n .
- Si el resultado es mayor a 9 entonces aplicar nuevamente el paso anterior (1) sobre el resultado.
- Si el resultado es igual a 9 entonces n es divisible por 9.
- Si el resultado es menor a 9 entonces n no es divisible por 9.

Escriba un programa para determinar si un número entero es divisible por 9, aplicando el criterio descrito anteriormente.

Ejercicio 22: Escriba un programa para calcular las siguientes sumatorias:

$$a) \sum_{i=0}^n \sum_{j=1}^m 1$$

$$b) \sum_{i=0}^n \sum_{j=1}^m i$$

$$c) \sum_{i=0}^n \sum_{j=1}^m (i + j)$$

$$d) \sum_{i=0}^n \sum_{j=i}^m (i + j)$$

Ejercicio 23: Escriba un programa dado dos números n y m naturales, genere la tabla de multiplicar para todos los números entre n y m . Por ejemplo, si $n=3$ y $m=5$ el programa deberá mostrar:

	1	2	3	4	5	6	7	8	9
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45

Ejercicio 24: 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:

- 1010
- 2020
- 3030
-
- 9090



Ejercicio 25: Todo número natural mayor a 1 puede ser expresado como un producto de números primos. Esta factorización es única. Por ejemplo, el número 60 puede ser descompuesto en factores primos $2 \times 2 \times 3 \times 5$. Observe que un mismo factor primo puede aparecer más de una vez en la factorización.

Realice un algoritmo y un programa en Pascal que solicite un número natural N y muestre su factorización.

Ejercicio 26: Nicómano de Gerasa descubrió la siguiente propiedad de los números naturales:

Al sumar el primer impar se obtiene el primer cubo ($1 \times 1 \times 1$): $1 = 1$
 Al sumar los dos siguientes impares se obtiene el segundo cubo ($2 \times 2 \times 2$): $3+5 = 8$
 Al sumar los tres siguientes impares se obtiene el tercer cubo ($3 \times 3 \times 3$): $7+9+11 = 27$
 Al sumar los cuatro siguientes impares se obtiene el cuarto cubo ($4 \times 4 \times 4$): $13+15+17+19 = 64$
 Etc...

Escriba un programa en Pascal para calcular y mostrar los cubos de los primeros N números naturales (con N ingresado por el usuario)



Ejercicio 27: Un atleta dispone de 3 meses de 4 semanas cada mes para prepararse para una competencia de maratón. Siempre recorre la misma ruta y sólo entrena 5 días a la semana. Ha registrado los tiempos que tardó durante los 3 meses y desea conocer el promedio por cada semana, por cada mes y total.

Los datos que se proporcionan son 60 valores reales correspondientes al tiempo del recorrido de cada día, y están almacenados en un archivo "tiempos.dat"

Escriba un programa en Pascal que muestre por pantalla la información que el atleta desea conocer.

DIA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	..
Tiempo	40.5	39.8	41.3	38.8	40.1	39.4	41.2	41.1	40.9	39.8	42.1	40.3	38.7	38.4	36.5	..
Semana	40.1					40.48					39.2					
Mes	39.8															