

Resolución de Problemas y Algoritmos

Clase 6 Repetición incondicional (sentencia FOR)



Blaise Pascal



Dr. Diego R. García



Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Bahía Blanca - Argentina

Repetición

Utilizando las primitivas “llenar” y “trasvasar”, escribir un algoritmo para obtener exactamente 4 litros de agua. Se dispone de un bidón de 30 litros vacío, 1 jarra graduada de 1 litro vacía, y una canilla.

Llenar jarra
 Trasvasar a bidón
 Llenar jarra
 Trasvasar a bidón
 Llenar jarra
 Trasvasar a bidón
 Llenar jarra
 Trasvasar a bidón

¿ y si fueran 14 lit. en el bidón?
¿ y 200 en un tanque de 500?



Repetir 4 veces:

Llenar jarra
Pasar a bidón



- **Repeticiones:** permiten especificar de una manera abreviada una secuencia repetida de operaciones.

Iteración: conceptos y vocabulario

- En un lenguaje de programación, las sentencias repetitivas permiten repetir una secuencia de sentencias un cierto número de veces.
- A la **repetición** también se la llama **iteración**.
- Al repetir (o iterar) un grupo de sentencias, este grupo vuelve a ejecutarse (en el mismo orden) una y otra vez.
- Es por esto que en la jerga informática, para hablar de una iteración también se usan las palabras **ciclo**, **lazo** o **bucle**. (En inglés: **loop**).

La secuencia de pasos:

"llenar jarra
pasar a bidón"

se ejecutará 4 veces.

Repetir 4 veces:

**Llenar jarra
Pasar a bidón**

Sentencia FOR

- En Pascal existen tres sentencias diferentes para realizar repetición.
- Una de ellas es la sentencia **FOR**
- En este ejemplo se repite 5 veces "writeln('hola')"

```
PROGRAM Muchos_saludos;
{Muestra 5 veces "hola" en pantalla}
VAR num: INTEGER;
BEGIN
writeln('Saludos:');
FOR num:= 1 TO 5
DO writeln('hola');
writeln(' pulse enter');
readln; // mantiene abierta consola
END.
```

**"Ciclo FOR" o
"bucle FOR."**
En inglés:
"FOR-DO Loop".



num
?
1
2
3
4
5


Sentencia FOR

- En este ejemplo se repite 5 veces "writeln(num);"

```

PROGRAM ListaNumeros;
{Muestra números enteros entre 1 y 5}
VAR num: INTEGER;
BEGIN
writeln('Números');
FOR num:= 1 TO 5
  DO writeln(num);
writeln(' pulse enter');
readln; // mantiene abierta consola
END.
                
```

Observe que se imprimirá el valor de num



num
?
1
2
3
4
5

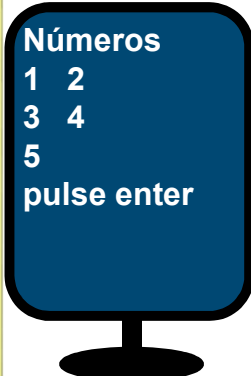
Resolución de Problemas y Algoritmos
Dr. Diego R. García
5

Sentencia FOR

- En este ejemplo se repite 5 veces una sentencia compuesta que es una secuencia de dos sentencias

```

PROGRAM De_a_dos;
{Muestra números enteros entre 1 y 5}
VAR num: INTEGER;
BEGIN
writeln('Números');
FOR num:= 1 TO 5 DO
  begin
  write(num, ' ');
  IF (num mod 2) = 0 then writeln
  end;
writeln(' pulse enter'); readln;
END.
                
```



num	(Num mod 2)=0
?	?
1	F
2	T
3	F
4	T
5	F

Resolución de Problemas y Algoritmos
Dr. Diego R. García
6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Concepto: repetición incondicional en Pascal

Sentencia FOR de Pascal:

La sentencia se ejecuta un número fijo de veces y luego continua en:

```
FOR V:= inicio TO fin
DO
  1 sentencia simple
  o compuesta ;
Otra sentencia siguiente;
```

Diagrama sintáctico FOR-TO:

Resolución de Problemas y Algoritmos Dr. Diego R. García 7

Concepto: repetición incondicional en Pascal

Las sentencias de un ciclo **FOR-TO** se ejecutan **CERO** o más veces dependiendo de las expresiones *inicio* y *fin* .

La sentencia se ejecuta un número fijo de veces y luego continua en:

```
FOR V:= inicio TO fin
DO
  1 sentencia simple
  o compuesta ;
Otra sentencia siguiente;
```

- **V** debe ser una **variable de tipo ordinal (char, integer o boolean)** a la que se suele llamar **variable de control**
- **inicio** y **fin** son expresiones cuyo valor resultante debe pertenecer al mismo tipo que la variable **V**.
- Al comenzar a **V** se le asigna el valor de **inicio**.
- Luego, **V** es **incrementada automáticamente de a uno** en cada repetición (hasta llegar al valor de **fin**).

Resolución de Problemas y Algoritmos Dr. Diego R. García 8

Conceptos: tipos ordinales en Pascal

Tipo de Dato: define el conjunto de valores posibles que puede tomar una variable, las operaciones que pueden aplicarse, y cual es la representación interna.

De los 4 tipos simples predefinidos que hemos visto, **INTEGER**, **CHAR** y **BOOLEAN** son **tipos ordinales** (**REAL** no es ordinal).

Los **tipos ordinales** poseen estas características:

- Tienen un orden. Tienen un primer y último elemento.
- Para cada elemento está definido el siguiente (a excepción del último) y el anterior (a excepción del primero).
- Tienen definidas las operaciones predefinidas:
 - **succ()** retorna el siguiente (excepto del último)
 - **pred()** retorna el anterior (excepto del primero)
 - **ord()** retorna el número de orden

Ejemplo

Escriba un programa para mostrar por pantalla todas las letras entre 'A' y 'E'.

```
PROGRAM ListaLetras;
{Muestra todos las letras entre 'A' y 'E'.}
VAR letra: char;
BEGIN
  writeln('Letras:');
  FOR letra:= 'A' TO 'E'
    DO writeln(letra);
  writeln('enter para continuar');
  readln; // mantiene abierta consola
END.
```



letra
A
B
C
D
E

Repetición incondicional

FOR $V := inicio$ **TO** fin
DO *sentencia simple o compuesta*

- *inicio* y *fin* son expresiones cuyo valor debe pertenecer al mismo tipo que la variable de control V .
- La *sentencia* (que puede ser compuesta), se repetirá un número fijo de veces: $(fin - inicio) + 1$.

FOR $V := 1$ **TO** $(2*2)+1$ **DO** `writeln(V);`

← repite 5 veces

- Si *fin* es menor estricto a *inicio* entonces se repetirá 0 veces.

FOR $V := 5$ **TO** 1 **DO** `writeln(V);`

← repite 0 veces

Resolución de Problemas y Algoritmos
Dr. Diego R. García
11

Repetición incondicional

- Al comenzar a V se le asigna el valor inicial y luego, V se incrementada automáticamente de a uno hasta llegar al valor final.

<p>FOR $V := 1$ TO 100 DO <code><sentencia></code></p>	<p>Aquí <code><sentencia></code> se repite... 100 veces: $100-1+1$</p>
<p>FOR $V := 100$ TO 199 DO <code><sentencia></code></p>	<p>Aquí <code><sentencia></code> se repite... 100 veces: $199-100+1$</p>
<p>FOR $V := -10$ TO -1 DO <code><sentencia></code></p>	<p>Aquí <code><sentencia></code> se repite... 10 veces: $-1 - (-10) + 1$</p>
<p>FOR $V := 1$ TO -2 DO <code><sentencia></code></p>	<p>Aquí <code><sentencia></code> se repite ... 0 veces</p>

Resolución de Problemas y Algoritmos
Dr. Diego R. García
12

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Sentencia FOR-DOWNTO

FOR V:= exp1 DOWNTO exp2 DO sentencia

- **exp1** y **exp2** pueden ser valores, variables, o expresiones del mismo tipo ordinal que la variable **V** (no puede ser tipo real).

FOR numero := 10 DOWNTO 0 DO write(numero);

FOR letra:= 'Z' DOWNTO 'A' DO write(letra);

Resolución de Problemas y Algoritmos
Dr. Diego R. García
13

Sentencia FOR-DOWNTO

FOR V:= exp1 DOWNTO exp2 DO sentencia

- El resultado de **exp1** y **exp2** deben poder calcularse justo antes de la ejecución del bucle FOR-DOWNTO. La variable **V** toma el valor inicial de evaluar **exp1** y luego, en cada iteración, **V** se **decrementa automáticamente de a uno** hasta llegar al valor de **exp2**.
- Si **val1** es el valor de **exp1** y **val2** el de **exp2**, entonces la **sentencia**, que puede ser compuesta, se repetirá **(val1 – val2 + 1)** veces.
- Si **val2** es mayor estricto a **val1** entonces se repetirá 0 veces.

FOR num:= 5 DOWNTO 1 DO write(num);

repite 5 veces

FOR num:= 1 DOWNTO 5 DO write(num);

repite 0 veces

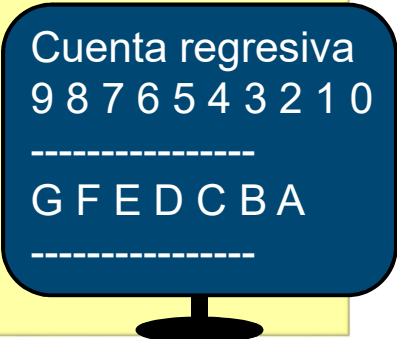
- Importante: La sentencia FOR incrementa (TO) o decrementa (DOWNTO) automáticamente de a un valor.

Resolución de Problemas y Algoritmos
Dr. Diego R. García
14

FOR-DOWNTO. Ejemplo.

```

PROGRAM CuentaRegresiva;
{Imprime en pantalla los dígitos de 9 a 0 en ese orden
 y las letras de G a A en ese orden}
VAR dig:integer; letra:char;
BEGIN
writeln('Cuenta regresiva');
FOR dig:=9 DOWNTO 0
 DO write(dig,' ');
writeln;
writeln('-----');
FOR letra:='G' DOWNTO 'A'
 DO write(letra,' ');
writeln;
writeln('-----');
END.
    
```



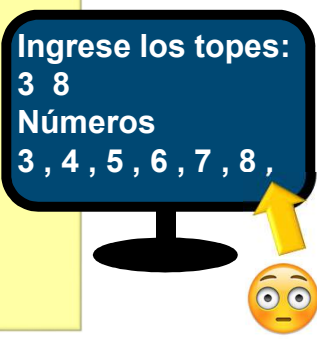
Resolución de Problemas y Algoritmos Dr. Diego R. García 15

Ejemplo

Escriba un programa para mostrar por pantalla todos los números enteros entre dos topes ingresados.

```

PROGRAM ListaNumeros;
{Muestra todos los números enteros desde
 tope inferior a tope superior}
VAR topeinf,topesup,num:INTEGER;
BEGIN
writeln('Ingrese los topes: ');
readln(topeinf, topesup);
writeln('Números');
FOR num:= topeinf TO topesup
 DO write(num,' ');
END.
    
```



topeinf	topesup	num
?	?	?
3	8	3
		4
		5
		6
		7
		8

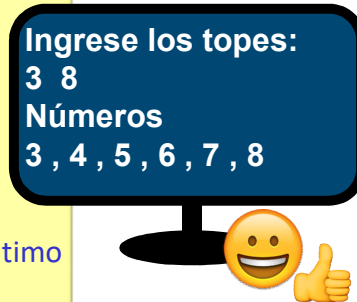
Resolución de Problemas y Algoritmos Dr. Diego R. García 16

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Ejemplo

Escriba un programa para mostrar por pantalla todos los números enteros entre dos topes ingresados.

```
PROGRAM ListaNumeros;
{Muestra todos los números enteros desde tope inferior a tope superior}
VAR topeinf,topesup,num:INTEGER;
BEGIN
  writeln('Ingrese los topes: ');
  readln(topeinf, topesup);
  writeln('Números');
  FOR num:= topeinf TO topesup -1
    DO write(num,' ');
  write(topesup) {Escribe por separado el último entero para evitar la coma final}
END.
```



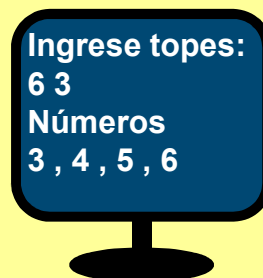
Ejemplo

Escriba un programa para mostrar por pantalla todos los números entre dos topes ingresados en cualquier orden.

```
PROGRAM ListaNumeros;
{muestra los números entre dos topes}
VAR topeinf,topesup,num,aux:INTEGER;
BEGIN
  writeln('Ingrese topes: '); readln(topeinf, topesup);
  IF topesup < topeinf {si los topes están invertidos}
  THEN begin {intercambio los valores de los topes}
    aux:=topeinf;
    topeinf:=topesup;
    topesup:=aux;
  end;
  writeln('Números');
  FOR num:= topeinf TO topesup -1
    DO write(num,' ');
  write(topesup)
END.
```

Casos de prueba:

3 6
6 3
6 6



aux	topeinf	topesup	num
?	?	?	?
?	6	3	?
6	3	6	?
			3
			4
			5
			6

Sentencia FOR-TO

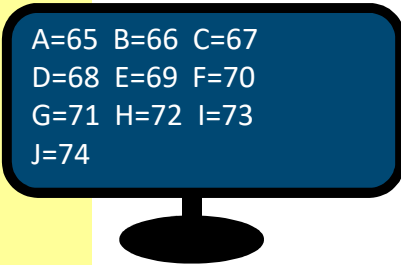
FOR V:= exp1 TO exp2 DO sentencia

- Tanto **exp1** como **exp2** pueden ser valores, variables, o expresiones siempre que sean del mismo tipo ordinal que **V** (no puede ser tipo real).
- Al comenzar a **V** se le asigna el valor de evaluar **exp1** y luego, **V** se incrementada automáticamente de a uno hasta llegar al valor de **exp2**.

```
N:=2;
FOR V := 1+1 TO N DO writeln(V);
FOR V := N+N TO sqr(N+N) DO writeln(V);
FOR V := N div 2 TO (N+10) - N + 2 DO writeln(V);
```

FOR-TO con tipo CHAR

```
PROGRAM letras_y_codigos;
{muestra en pantalla los códigos
ASCII de algunas letras en columnas}
CONST ultima='J'; columnas=3;
VAR letra: char; contador: integer;
BEGIN
  contador:=1;
  FOR letra:='A' TO ultima DO
  BEGIN {comienzo del ciclo for}
    write(letra,'=', ord(letra),' ');
    {baja de renglón cada "columnas" veces}
    if contador mod columnas = 0
    then writeln;
    contador:=contador + 1;
  END {fin del ciclo FOR}
END.
```



A=65 B=66 C=67
D=68 E=69 F=70
G=71 H=72 I=73
J=74

Comienza con el valor 'A' y luego incrementa automáticamente de a uno pasando por todos los valores del código ASCII hasta llegar al valor final ('J')

Problema Propuesto

Escriba un programa que calcule el promedio de una cantidad conocida (e ingresada por el usuario) de números reales.

Ingrese la cantidad de valores:
4
Ingrese un valor: 8.2
Ingrese un valor: 0.2
Ingrese un valor: -3.0
Ingrese un valor: 5.2
El promedio es: 2.65

Resolución de Problemas y Algoritmos
Dr. Diego R. García
21

Problema propuesto

Escriba un programa que calcule el promedio de una cantidad conocida (e ingresada por el usuario) de números reales.

Solución: El promedio consiste de sumar todos los números reales ingresados y dividir por la cantidad ingresada.

La dificultad que presenta no conocer hasta el momento de la ejecución cuántos valores hay que sumar, se resuelve calculando la suma a medida que los valores son ingresados (sin guardar los valores individuales)

Algoritmo "promedio":

Leer cantidad

Suma ← 0

Repetir cantidad veces

leer valor

suma ← valor + suma

Mostrar suma/cantidad

La flecha "←" es el símbolo de asignación usualmente usado en algoritmos.

El dato "suma" (inicialmente en cero) va acumulando el resultado de sumar los valores leídos.

Resolución de Problemas y Algoritmos
Dr. Diego R. García
22

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Un programa posible para promedio

```

PROGRAM Promedio;
{Calcula el promedio de una cantidad fija de valores
reales ingresados por el usuario }
VAR cantidad, control: integer;
    suma, valor, prom: real;
BEGIN
  writeln('Cantidad de valores: '); readln(cantidad);
  suma:=0; // valor inicial (neutro para la suma)
  FOR control:= 1 TO cantidad DO
    begin
      writeln('Ingrese un valor: ');
      readln(valor); // cada valor leído borra el anterior
      suma:=suma + valor; // pero la suma se acumula
    end; // en este punto suma tiene acumulado
  prom:= suma/cantidad;
  writeln('El promedio es: ',prom:0:2);
END.
        
```

Traza

cantidad	suma	valor	control	prom
?	?	?	?	?
3	0	?	?	?
	8	8	1	?
	15	7	2	?
	24	9	3	8

Resolución de Problemas y Algoritmos
Dr. Diego R. García
23

Observaciones sobre repetición incondicional

FOR V:= Exp1 TO (DOWNTO) Exp2 DO sentencia

- Tanto en un ciclo FOR-TO como en FOR-DOWNTO, el valor de *Exp1* (llamado *valor inicial*) y el de *Exp2* (llamado *valor final*) deben poder calcularse al comenzar la repetición, de lo contrario es un error de programación.

```

PROGRAM Incorrecto;
{¿Por qué es incorrecto? }
VAR v, inicio, tope: integer;
BEGIN
  tope:=10;
  FOR v := inicio TO tope
  DO writeln (v);
END.
        
```

v	inicio	tope
?	?	?
		10


MAL

**Error de programación:
inicio sin valor**

Resolución de Problemas y Algoritmos
Dr. Diego R. García
24

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Muy importante



**FOR V:= <expresión1> TO (DOWNTO) <expresión2>
DO <Bloque de Sentencias>**

Ya sea para un ciclo FOR-TO o FOR-DOWNTO, en RPA será considerado error de programación:


- 1) Cambiar el valor de la variable de control V, dentro del bloque de sentencias de un ciclo FOR.
- 2) Cambiar el valor de cualquier variable de <expresión1> o <expresión2>, dentro del bloque de sentencias de un FOR, con motivo de que cambien los límites de la iteración.

Si surge la necesidad de hacerlo es porque tendría usar una repetición condicional con **REPEAT** o **WHILE** (que veremos muy pronto...).

Resolución de Problemas y Algoritmos Dr. Diego R. García 25


```
...
FOR V:= 1 TO 100
DO begin
  writeln(V);
  V:= V + 5;
end;
...
```

```
...
FOR V:= 1 TO 100 DO
begin
  writeln(V);
  if V=12 then V:= 100;
end;
...
```



Mal **Mal**

Error de programación



Importante: es un error de programación intentar controlar "manualmente" la variable de control o los límites de un for.

Lazarus dice: *Error: Illegal assignment to for-loop variable "v"*
Traducido: Error: asignación ilegal a la variable de control "v"


Resolución de Problemas y Algoritmos Dr. Diego R. García 26

Error de programación


```


...
ultimo:= 100;
FOR V:= 1 TO ultimo DO
begin
  writeln(V);
  if V = 12 then ultimo:=13;
end;
...
        
```

Es un error pensar que cambiando el valor del límite, la cantidad de repeticiones de un FOR puede cambiar o dejar de repetir.



Importante: es un error de programación intentar controlar “manualmente” la variable de control o los límites de un FOR.





Resolución de Problemas y Algoritmos
Dr. Diego R. García
27

Repeticiones anidadas

```
FOR v:= 1 TO 3 DO
  writeln( v );
```

¿cuántas veces se ejecuta `writeln(v)`?

```
FOR v:= 1 TO 3 DO
  FOR h:= 1 TO 2 DO
    writeln( v, h);
```

¿cuántas veces se ejecuta `writeln(v,h)`?

Obs: usan diferentes variables de control
¿Por qué?

```
FOR v:= 1 TO 3 DO
  FOR h:= 1 TO 2 DO
    FOR t:= 5 downto 1
      DO writeln(v, h, t);
```

¿cuántas veces se ejecuta `writeln(v,h,t)`?

Resolución de Problemas y Algoritmos
Dr. Diego R. García
28

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Usos permitidos de la variable de control

- La variable de control si puede usarse y modificarse en otras partes del programa que estén “afuera” del ciclo FOR.
- Como muestra el ejemplo 1, si es posible cambiar la variable de control V antes o después del ciclo FOR.
- En el ejemplo 2 se muestra que la misma variable de control V también usarse como variable de control para otro ciclo FOR siempre que uno no esté anidado en el otro.

{ejemplo 1: válido}

```
V:= 9;
writeln(V);
FOR V:= 1 TO 3
DO writeln(V);
V:= 8; writeln(V);
```

{ej 2: válido}

```
FOR V:= 1 TO 3
DO writeln(V);
FOR V:= 4 TO 9
DO writeln(V);
V:= 4; writeln(V);
```

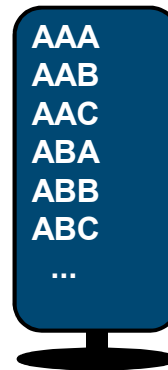
{ej 3: inválido}

```
FOR V:= 1 TO 3 DO
begin
FOR V:= 4 TO 9
DO writeln(V);
end;
```



Repeticiones anidadas: ejemplo

```
PROGRAM combinaLetras;
{muestra las 27 combinaciones
de las letras A B y C}
VAR L1,L2,L3: CHAR;
BEGIN
FOR L1 := 'A' TO 'C' DO
FOR L2 := 'A' TO 'C' DO
FOR L3 := 'A' TO 'C' DO
writeln (L1,L2,L3)
END.
```



L1	L2	L3
?	?	?
A	A	A
A	A	B
A	A	C
A	B	A

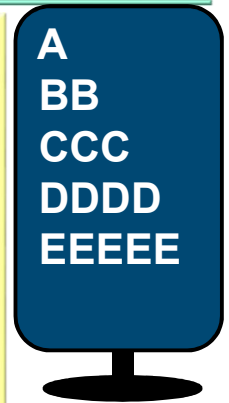
- **Problema propuesto:** Un dominio automotor (patente) es una combinación de letras y dígitos. Escriba un programa que muestre TODAS las patentes posibles. ¿Cuántas son?

Sentencias FOR-TO anidadas. Ejemplo.

```

PROGRAM FOR_anidados;
{ejemplo de bucles FOR anidados donde el ciclo interno
tiene un límite (cant) que va cambiando en cada iteración}
VAR Letra: char; i,cant: integer;
BEGIN
cant:=1; {cantidad de letras por renglón}
FOR Letra := 'A' TO 'E'
DO
    BEGIN {imprime un renglón de "Letra"s}
        FOR i := 1 TO cant DO write(Letra);
        writeln; {baja de renglón}
        cant:=cant+1;{incrementa la cantidad por renglón}
    END;
END.
    
```

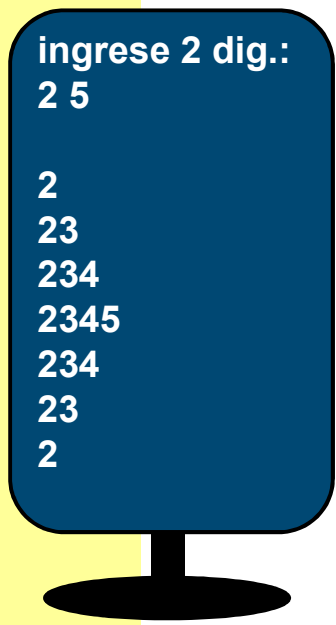
Realice la traza.



En este lugar del programa, es perfectamente lícito modificar la variable "cant", ya que el cambio se produce fuera del ciclo FOR que la usa como límite superior.

```

PROGRAM FOR_anidados2;
VAR cant, num, pri, ult : integer;
BEGIN {otro ejemplo de límites que cambian}
writeln('ingrese 2 dig. '); readln(pri, ult);
{primera mitad incrementando}
FOR cant := pri TO ult DO
    BEGIN
        FOR num := pri TO cant DO write(num);
        writeln;
    END;
{segunda mitad decrementando}
FOR cant := ult-1 DOWNTO pri DO
    BEGIN
        FOR num := pri TO cant DO write(num);
        writeln;
    END;
END.
    
```



El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Problema para practicar

- **Problema propuesto:** Escriba un programa que pida al usuario un valor $0 < N < 10$ y dibuje una forma como la que sigue (por ejemplo para $N=4$):

ingrese N: 4

1

22

333

4444

333

22

1

1. Entender el problema
2. Buscar solución:
3. Buscar ejemplos particulares
4. Dividir el problema en partes
5. Escribir algoritmo
6. Verificar con una traza.
7. Escribir programa
8. Verificar con una traza.

Resolución de Problemas y Algoritmos Dr. Diego R. García 33

Problema propuesto

- Escriba un programa para calcular un número natural elevado a una potencia (también natural).
- **Ejemplos:** $2^3=8$ $3^2=9$ $1^6=1$ $2^{10}=1024$
 $2^3=2*2*2=8$ $3^0=1$ $3^2=3*3=9$ $1^6=1*1*1*1*1*1=1$
- **Solución:** multiplicar “base” “exponente” veces

Algoritmo “potencia”:

Leer base y exponente

Potencia $\leftarrow 1$

Repetir exponente veces

potencia \leftarrow potencia * base

Mostrar potencia

Resolución de Problemas y Algoritmos Dr. Diego R. García 34

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Un posible programa para "potencia"

```

PROGRAM CalculoPotencia;
{Calcula un número natural elevado a una potencia
(también natural) }
VAR base, exponente, potencia, aux: integer;
BEGIN
writeln('Ingrese base y exp : ');
readln(base, exponente);
potencia := 1;
FOR aux := 1 TO exponente
    DO potencia:=potencia * base;
write(base, ' a la ', exponente);
writeln(' es ', potencia);
END.
                
```

Base	exponente	aux	potencia
?	?	?	?
2	7	1	2
		2	4
		3	8
		4	16
		5	32
		6	64
		7	128

Ingrese base y exp:
 2 7
 2 a la 7 es 128

Resolución de Problemas y Algoritmos

Dr. Diego R. García

35

Problema propuesto

Factorial de un número N se denota con N!
 y se define: $N! = 1 * 2 * 3 * 4 * \dots * N$
 $0! = 1$

Ejemplos:
 $1! = 1$
 $2! = 2$ $3! = 6$ $4! = 24$ $5! = 120$
 $6! = 720$
 $7! = 5.040$
 $8! = 40.320$
 $9! = 362.880$
 $10! = 3.628.800$

Escriba un programa para calcular el factorial de un número ingresado por el usuario.

Resolución de Problemas y Algoritmos

Dr. Diego R. García

36

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Algoritmo propuesto

Problema: Escriba un programa para calcular el Factorial de un número N ingresado por el usuario.

Como $N! = 1 * 2 * 3 * 4 * \dots * N$
 Por lo tanto tengo que hacer N multiplicaciones

Algoritmo factorial:
 Leer N
 factorial \leftarrow 1
 factor \leftarrow 1
 repetir N veces:
 factorial \leftarrow factorial * factor
 factor \leftarrow factor + 1
 Mostrar factorial en pantalla

Programa para "factorial"

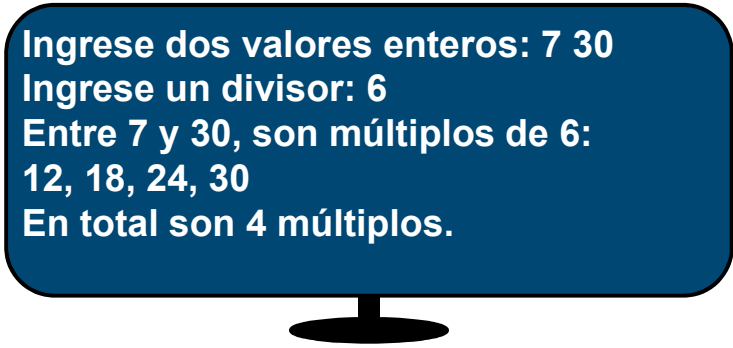
```
PROGRAM CalcularFactorial;
{ calcula factorial de un número ingresado}
VAR numero, factor, factorial: INTEGER;
BEGIN
writeln('ingrese número >= 0');
readln(numero);
factorial:=1;
FOR factor:=1 TO numero
DO factorial:=factorial * factor;
Writeln(' El factorial de ',numero, ' es ', factorial);
END.
```

numero	factor	factorial
3	3	3
5	1	1
	2	2
	3	6
	4	24
	5	120

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:
 "Resolución de Problemas y Algoritmos. Notas de Clase". Diego R. García. Universidad Nacional del Sur. (c) 03/09/2019.

Problemas para practicar

Escriba un programa que dados tres valores enteros V1, V2 y N ingresados por el usuario, muestre y cuente cuantos enteros hay entre V1 y V2 que sean múltiplos de N.



Ingrese dos valores enteros: 7 30
Ingrese un divisor: 6
Entre 7 y 30, son múltiplos de 6:
12, 18, 24, 30
En total son 4 múltiplos.

Información adicional

Blaise Pascal (1623 -1662)

Matemático, físico, filósofo y teólogo francés, considerado uno de los padres de las computadoras junto con *Charles Babbage*.

Desde los 12 se dedicó a las matemáticas, cuando tenía 19 años comenzó a trabajar en su calculadora mecánica (llamada Pascaline o Arithmetique).

Luego se dedicó al física en especial la presión atmosférica y posteriormente a la filosofía.

Murió en Paris a los 39 años.

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



Roue Pascaline

En 1640, su padre fue nombrado Comisario Real y jefe de recaudación de impuestos para la Normandía con asiento en Ruan.

Allí, en 1642, Pascal inventó la Roue Pascaline, «Rueda de Pascal» o Pascalina. Pretendía ayudar a su padre buscando crear un dispositivo que pudiera reducir un poco su carga de trabajo.

Inicialmente solo permitía realizar adiciones, pero recibió permanentes mejoras, siendo finalmente capaz de realizar restas. Pascal la hizo patentar, pero no se cumplieron sus expectativas de hacerse rico. Las máquinas, confeccionadas una a una y a mano, eran demasiado caras y solo llegó a fabricar 50, de las que subsisten nueve.

