

Resolución de Problemas y Algoritmos

Clase 6



Dr. Alejandro J. García
<http://cs.uns.edu.ar/~ajg>



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

Repaso: Repetición incondicional

FOR V:= *valor_inicial* **TO** *valor_final*
DO *sentencia (simple o compuesta)*

- *valor_inicial* y *valor_final* son expresiones cuyo valor debe pertenecer al mismo tipo que V.
- La *sentencia* (que puede ser compuesta), se repetirá (*valor_final* – *valor_inicial* + 1) veces.
- Si *valor_final* es menor estricto a *valor_inicial* entonces se repetirá 0 veces.
- La *variable* V se suele llamar variable de control.
- Al comenzar a V se le asigna *valor_inicial*.
- Luego, V es **incrementada automáticamente de a uno** en cada repetición (hasta llegar a *valor_final*).

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 2

```

PROGRAM CalcularFactorial;
    { calcula factorial de un número leído }
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);
    Readln; {espera que presione enter ....}
END.
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 3

Problema propuesto

Problema: Escribir un programa para ver si un número N **es primo** (*Definición: un número es primo si es un entero positivo mayor que 1, que es divisible solamente por si mismo y la unidad*)

Ejemplos: 2, 3, 7, 11 y 2003: son primos
 4 no es primo es divisible por 2
 2001 no es primo, es divisible por 3
 121 no es primo, es divisible por 11

Una solución: cuento la cantidad de divisores que tiene N, si tiene 0, entonces es primo, si tiene un divisor (o más) entonces no es primo.

Tarea: escriba el algoritmo y el programa.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 4

Sentencia FOR-DOWNTO

La sentencia **FOR-DOWNTO** es análoga al **FOR-TO**, con la diferencia de que el valor de la variable de control se **decrementa de a uno** automáticamente en cada iteración.

FOR variable := val-inicial **DOWNTO** val-final
DO *sentencia*

FOR numero := 100 **DOWNTO** 90 **DO** write(numero);

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 5

Sentencia FOR-DOWNTO

FOR V:= *valor_inicial* **DOWNTO** *valor_final*
DO *sentencia (simple o compuesta)*

- Al comenzar a V se le asigna *valor_inicial*.
- Luego, V es **decrementada automáticamente de a uno** en cada repetición (hasta llegar a *valor_final*).
- *valor_inicial* y *valor_final* son expresiones cuyo resultado debe pertenecer al mismo tipo que V.
- La *sentencia*, que puede ser compuesta, se repetirá (*valor_inicial* – *valor_final* + 1) veces.
- Si *valor_final* es mayor estricto a *valor_inicial* entonces se repetirá 0 veces.

Resolución de Problemas y Algoritmos Dr. Alejandro J. 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”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

FOR-DOWNTO. Ejemplo.

```

PROGRAM CuentaRegresiva;
{Este programa muestra los números de 9 a 0}
VAR j:INTEGER;
BEGIN
  writeln('Cuenta regresiva');
  FOR j:=9 DOWNTO 0
  DO write(j:2);
  writeln;
  writeln('Fin!!');
END.
    
```

Cuenta regresiva
9 8 7 6 5 4 3 2 1 0
Fin!!

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 7

FOR es una repetición **incondicional**. El valor inicial y el final de V no debe cambiar durante la ejecución del bloque de sentencias. Si surge la necesidad de hacerlo es porque **DEBE** usar **REPEAT** o **WHILE**.

```

FOR V:= <expresión1> to/downto <expresión2>
DO <Bloque de Sentencias>
    
```

≠

En RPA será considerado error de programación:

- 1) Cambiar el valor de la variable de control V, dentro del bloque de sentencias de un 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.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 8

```

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

```

V:= 5; writeln(V);
FOR V:= 3 DOWNTO 1
DO writeln(V);
V:= 4; writeln(V);
        
```

- Al comenzar a V se le asigna *valor_inicial* y luego, **incrementada o decrementada automáticamente de a uno** hasta llegar a *valor_final*.

≠

Importante: No debe cambiarse el valor de la variable de control V dentro del bloque de sentencias del FOR. Es considerado ERROR de programación.

Obs: Si es posible cambiar V antes o después del ciclo FOR.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 9

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?

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

```

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

Sentencias FOR-TO anidadas. Ejemplo.

```

PROGRAM letras;
VAR Letra: CHAR; i: integer;
BEGIN
  FOR Letra := 'A' TO 'E'
  DO BEGIN
    FOR i := 1 TO (ord(Letra) - ord('A')) + 1
    DO write(Letra);
    writeln;
  END;
END.
    
```

A
BB
CCC
DDDD
EEEE

Realice la traza.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 11

Sentencias FOR-TO anidadas. Ejemplo.

```

PROGRAM anidados;
VAR cant, num, pri, ult : integer;
BEGIN
  writeln('ingrese 2 dig. ');
  readln(pri, ult);
  FOR cant := pri TO ult DO
  BEGIN
    FOR num := pri TO cant DO write(num);
    writeln;
  END;
  FOR cant := ult-1 DOWNTO pri DO
  BEGIN
    FOR num := pri TO cant DO write(num);
    writeln;
  END;
END.
    
```

ingrese 2 dig.:
2 5

2
23
234
2345
234
23
2

Resolución de Problemas y Algoritmos Dr. Alejandro J. 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”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

```
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;
```

- **Problema propuesto:** Un dominio automotor (patente) es una combinación de 3 letras y 3 dígitos. Escriba un programa que muestre TODAS las patentes posibles. ¿Cuántas son?

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 13

Palabras Reservadas

Las **palabras reservadas** son aquellas que ya tienen un significado en el lenguaje Pascal, y el programador sólo puede usarlas con ese significado.

and	end	nil	set
array	file	not	then
begin	for	of	to
case	function	or	type
const	goto	packed	until
div	if	procedure	var
do	in	program	while
downto	label	record	with
else	mod	repeat	

No son reservadas
Write, read, real, char, maxint

Son predefinidas
¿Es posible?
const div = 12;
const write = 1;

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

Problemas para practicar

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 15

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.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 16

Problemas 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 $N=4$):

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 17

Problemas para practicar

- **Problemas propuestos:** Escriba programas que pidan al usuario un valor $0 < N < 10$ y dibujen formas como las que siguen (por ejemplo para $N=4$):

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 18

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”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.