



Resolución de Problemas y Algoritmos

Clase 6 Repeticiones anidadas



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

Palabras Reservadas en Pascal

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;

19:18 Resolución de Problemas y Algoritmos
Dr. Alejandro J. García
2

Problemas para practicar

- 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: 2.8
 El promedio es: 2.05

19:18 Resolución de Problemas y Algoritmos
Dr. Alejandro J. García
3

Repaso: Repetición incondicional

FOR V:= Exp1 TO Exp2 DO sentencia

- Exp1* y *Exp2* son expresiones cuyo valor debe pertenecer al mismo tipo que *V* (variable de control).
- Al comenzar a *V* se le asigna el valor de *Exp1*.
- Luego, *V* es **incrementada automáticamente de a uno** en cada repetición (hasta el valor de *Exp2*).
- El valor de *Exp1* (*valor inicial*) y el de *Exp2* (*valor final*) deben poder calcularse al comenzar la repetición, de lo contrario es un error de programación (vea ejemplos en el pizarrón)
- La *sentencia* se repetirá (*valor final* - *valor inicial* + 1) veces, o 0 veces si *valor final* < *valor inicial*

19:18 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);

19:18 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.

19:18 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)1998-2013.

FOR-DOWNTO. Ejemplo.

```

PROGRAM CuentaRegresiva;
VAR dig:integer; letra:char;
BEGIN
writeln('Cuenta regresiva');
FOR dig:=9 DOWNTO 0
DO write(dig);
writeln('-----');
FOR letra:='G' DOWNTO 'A'
DO write(letra);
writeln('-----');
END.
    
```

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

G F E D C B A

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. García 7

```

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.

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. García 8

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.

19:18 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?

```

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

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

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. García 10

Sentencias FOR-TO anidadas. Ejemplo.

```

PROGRAM letras;
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.
    
```

A
BB
CCC
DDDD
EEEE

Realice la traza.

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. García 11

```

PROGRAM anidados;
VAR cant, num, pri, ult : integer;
BEGIN
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.
    
```

ingrese 2 dig.:
2 5

2
23
234
2345
234
23
2

19:18 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)1998-2013.

```

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?

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. García 13

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.

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

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.

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. García 15

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 $N=4$):

19:18 Resolución de Problemas y Algoritmos Dr. Alejandro J. 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”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2013.