



Resolución de Problemas y Algoritmos


Clase 6 Repetición incondicional (sentencia FOR)



Blaise Pascal



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

Conceptos de las clases anteriores

¿Preguntas?

- Algoritmo. Primitiva. Traza.
- Computadora. Hardware. Bit. Byte. Memoria RAM.
- Lenguaje de programación. Programa. Código fuente.
- Compilador. Código ejecutable.
- Sintaxis de un lenguaje. Diagrama sintáctico.
- Pascal:
 - Identificadores reservados y predefinidos
 - Constantes, variables y tipos de datos.
 - Primitivas: asignación (:=) read, readln, write, writeln
 - Tipos predefinidos: real, integer, char, boolean.
 - Expresiones. Operaciones y funciones predefinidas.
 - Expresiones lógicas. Expresiones equivalentes.
 - Sentencia condicional IF-THEN-ELSE

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

Unidades de almacenamiento

Unidad	Equivale a
1 Bit	
1 Byte	8 bits
1 Kilobyte (KB)	1000 bytes
1 Megabyte (MB)	1000 KB = 1.000.000 bytes
1 Gigabyte (GB)	1000 MB = 1.000.000.000 bytes
1 Terabyte (TB)	1000 GB = 1.000.000.000.000 bytes
1 Petabyte (PB)	1000 TB = 1.000.000.000.000.000 bytes
1 Exabyte (EB)	1000 PB = 1.000.000.000.000.000.000 bytes
1 Zettabyte (ZB)	1000 EB = 1.000.000.000.000.000.000.000 bytes
1 Yottabyte (YB)	1000 ZB = 1.000.000.000.000.000.000.000.000 bytes

[Más detalles en https://en.wikipedia.org/wiki/Binary_prefix](https://en.wikipedia.org/wiki/Binary_prefix)

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

Ejemplos

Unidad	
Bit	• El código fuente en Pascal "prueba.pas" de la clase pasada ocupa 134 bytes .
Byte	• El ejecutable de ese programa ocupa 69 KB .
Kilobyte (KB)	• El archivo pdf de la clase pasada ocupa 1,3 MB .
Megabyte (MB)	• La memoria RAM de una PC en 2015 era de 4 GB .
Gigabyte (GB)	• En 2015 un disco rígido típico era de 1 TB .
Terabyte (TB)	• El contenido de la web en 1995 se estimaba en 8 PB .
Petabyte (PB)	• El contenido de la web en 2009 se estimaba en 500 EB .
Exabyte (EB)	• En un gramo de ADN contiene unos 455 EB de información.
Zettabyte (ZB)	• El contenido de la web en 2013 se estimaba en 4 ZB (¿unos 8 gramos de ADN?).
Yottabyte (YB)	¿Más curiosidad? vea: https://en.wikipedia.org/wiki/Yottabyte

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

Conceptos y vocabulario

- En un lenguaje de programación, las sentencias repetitivas permiten repetir un cierto número de veces un grupo (bloque) de sentencias.
- A la **repetición** también se la llama **iteración**.
- Al repetir (o iterar) un grupo de sentencias, este grupo vuelve a comenzar 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.*

Repetir 10 veces:
Llenar botella
Pasarse a bidón

El grupo:
"Llenar botella
pasar a bidón"
se ejecutará en secuencia
10 veces.

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

Conceptos y vocabulario

- En **Pascal existen tres sentencias diferentes** para realizar repetición.
- Por ejemplo: en la siguiente iteración, la secuencia de sentencias (sentencia compuesta) dentro del ciclo FOR se ejecutará 10 veces.

```
FOR V:= 1 TO 10 DO
begin
... secuencia de sentencias...
end;
```

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

Repetir 10 veces:
Llenar botella
Pasarse a bidón

El grupo:
"Llenar botella
pasar a bidón"
se ejecutará en secuencia
10 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)2016.

Concepto: repetición incondicional en Pascal

Sentencia FOR de Pascal:

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

FOR V:= valor_inicial TO valor_final DO
1 sentencia simple o compuesta ;
Otra sentencia siguiente;

Diagrama sintáctico FOR-TO:

```

    sentencia FOR → (for) → variable → := → expresión
                               |
                               ↓
                               (to) → expresión → do → proposición
    
```

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

Concepto: repetición incondicional en Pascal

Las sentencias de un ciclo **FOR-TO** se ejecutan **CERO** o más veces dependiendo de **valor_inicial** y **valor_final**.

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

FOR V:= valor_inicial TO valor_final DO
1 sentencia simple o compuesta ;
Otra sentencia siguiente;

- V debe ser una **variable de tipo ordinal** (que se suele llamar **variable de control**)
- valor_inicial** y **valor_final** son expresiones cuyo valor resultante debe pertenecer al mismo tipo que la variable **V**.
- 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 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

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

Ejemplo (p)

Escriba un programa para mostrar por pantalla todos los números enteros entre 1 y 5.

```

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

num	?
	1
	2
	3
	4
	5

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 10

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 la variable de control **V**.
- La **sentencia** (que puede ser compuesta), se repetirá un número fijo de veces: **valor_final - valor_inicial + 1**.

FOR V:= 1 TO (2*2)+1 DO writeln(V); ← repite 5 veces

- Si **valor_final es menor estricto** a **valor_inicial** entonces se repetirá 0 veces.

FOR V:= 5 TO 1 DO writeln(V); ← repite 0 veces

Resolución de Problemas y Algoritmos Dr. Alejandro J. 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.

FOR V:= 1 TO 100 DO <sentencia> → Aquí **<sentencia>** se repite 100 veces: $100-1+1$

FOR V:= 100 TO 199 DO <sentencia> → Aquí **<sentencia>** se repite 100 veces: $199-100+1$

FOR V:= -10 TO -1 DO <sentencia> → Aquí **<sentencia>** se repite 10 veces: $-1 - (-10) + 1$

FOR V:= 1 TO -2 DO <sentencia> → Aquí **<sentencia>** se repite 0 veces

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)2016.

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. Alejandro J. 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 o decrementa automáticamente de a un valor. Si se quiere que sea de a más de uno, debe programarse "a mano".

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

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	toesup	num
3	8	3
		4
		5
		6
		7
		8

Ingrese los topes:
 3 8
 Números
 3, 4, 5, 6, 7, 8

Resolución de Problemas y Algoritmos Dr. Alejandro J. 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 -1
  DO write(num,' ');
  write(topesup) {Escribe por separado el último entero para evitar la coma final}
END.
    
```

Ingrese los topes:
 3 8
 Números
 3, 4, 5, 6, 7, 8

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

Ejemplo

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

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

```

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.
    
```

Ingrese topes:
 6 3
 Números
 3, 4, 5, 6

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

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 incrementa **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 sqrt(N+N) DO writeln(V);
FOR V := N div 2 TO (N+10) - N + 2 DO writeln(V);

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)2016.

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')

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 19

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.
    
```

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

G F E D C B A

dig	letra
?	?

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 20

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. Alejandro J. 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 todo los números reales ingresados y dividir por la cantidad ingresada.

La dificultad que presenta no conocer hasta el momento de la ejecución cuantos 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. Alejandro J. García 22

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;
  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. Alejandro J. García 23

Problema propuesto

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

Algoritmo "potencia":
Leer base y exponente
Potencia ← 1
Repetir exponente veces
 potencia ← potencia * base
Mostrar potencia

Resolución de Problemas y Algoritmos Dr. Alejandro J. 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". Alejandro J. García. Universidad Nacional del Sur. (c)2016.

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. Alejandro J. García 25

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. Alejandro J. García 26

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 27

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
?	?	?
5	1	1
	2	2
	3	6
	4	24
	5	120

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 28

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.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 29

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. Alejandro J. García 30

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)2016.

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. Alejandro J. García 31

```
...
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;
...
```

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. Alejandro J. García 32

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. Alejandro J. García 33

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. Alejandro J. García 34

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.
END.
```

AAA
AAB
AAC
ABA
ABB
ABC
...

	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?

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 35

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.
```

A
BB
CCC
DDDD
EEEE

Realice la traza.

Resolución de Problemas y Algoritmos Dr. Alejandro J. 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". Alejandro J. García. Universidad Nacional del Sur. (c)2016.

```
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.
```

ingrese 2 dig.:
2 5

2
23
234
2345
234
23
2

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 37

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. Alejandro J. García 38

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.

<p><i>{ejemplo 1: válido}</i></p> <pre>V:= 9; writeln(V); FOR V:= 1 TO 3 DO writeln(V); V:= 8; writeln(V);</pre>	<p><i>{ej 2: válido}</i></p> <pre>FOR V:= 1 TO 3 DO writeln(V); FOR V:= 4 TO 9 DO writeln(V); V:= 4; writeln(V);</pre>	<p><i>{ej 3: inválido}</i></p> <pre>FOR V:= 1 TO 3 DO begin FOR V:= 4 TO 9 DO writeln(V); end;</pre>
--	--	--

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 39

Información adicional

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 40

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

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 41

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.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 42

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)2016.