

Soluciones a los Ejercicios Propuestos en el libro:

EMPEZAR DE CERO A PROGRAMAR EN

lenguaje C

Autor: Carlos Javier Pes Rivas (correo@carlospes.com)

Capítulo 13

INSTRUCCIONES DE CONTROL REPETITIVAS

EJERCICIOS PROPUESTOS

Bucle mientras

13.1	Área de un cubo.....	2
13.2	Áreas de triángulos.....	3
13.3	Traza y salida por pantalla (bucle mientras).....	5

Bucle hacer...mientras

13.4	Suma de los cuadrados calculados.....	6
13.5	Suma y cantidad de resultados calculados.....	8
13.6	Traza y salida por pantalla (bucle hacer...mientras).....	10

Bucle para

13.7	Sucesión de números 1 -1 2 -2 3 -3 4 -4 5 -5.....	11
13.8	Primeras diez potencias de 2.....	12
13.9	Números impares del -9 al 3.....	12
13.10	Números pares del -50 al -80.....	13
13.11	Números impares del -9 al 3.....	14
13.12	Traza y salida por pantalla (bucle para).....	14

Anidamiento

13.13	Suma de números pares e impares.....	16
13.14	Ceros introducidos.....	18
13.15	Número mayor introducido y su posición.....	22
13.16	Número mayor y menor introducidos.....	26
13.17	Traza y salida por pantalla (bucle hacer...mientras en mientras).....	30

Miscelánea

13.18	¿Cuántos números son primos?.....	32
13.19	Número inverso.....	34
13.20	Lanzar un dado hasta sumar 21 ó más.....	36
13.21	Gastos de un monedero.....	38
13.22	Tiempo transcurrido entre dos horas.....	39
13.23	Desglose de moneda.....	42
13.24	Mostrar tres números en orden creciente.....	47
13.25	Rellenar espacios en blanco.....	49



2 Empezar de cero a programar en lenguaje C

EJERCICIO PROPUESTO 13.1

Área de un cubo

Solución:

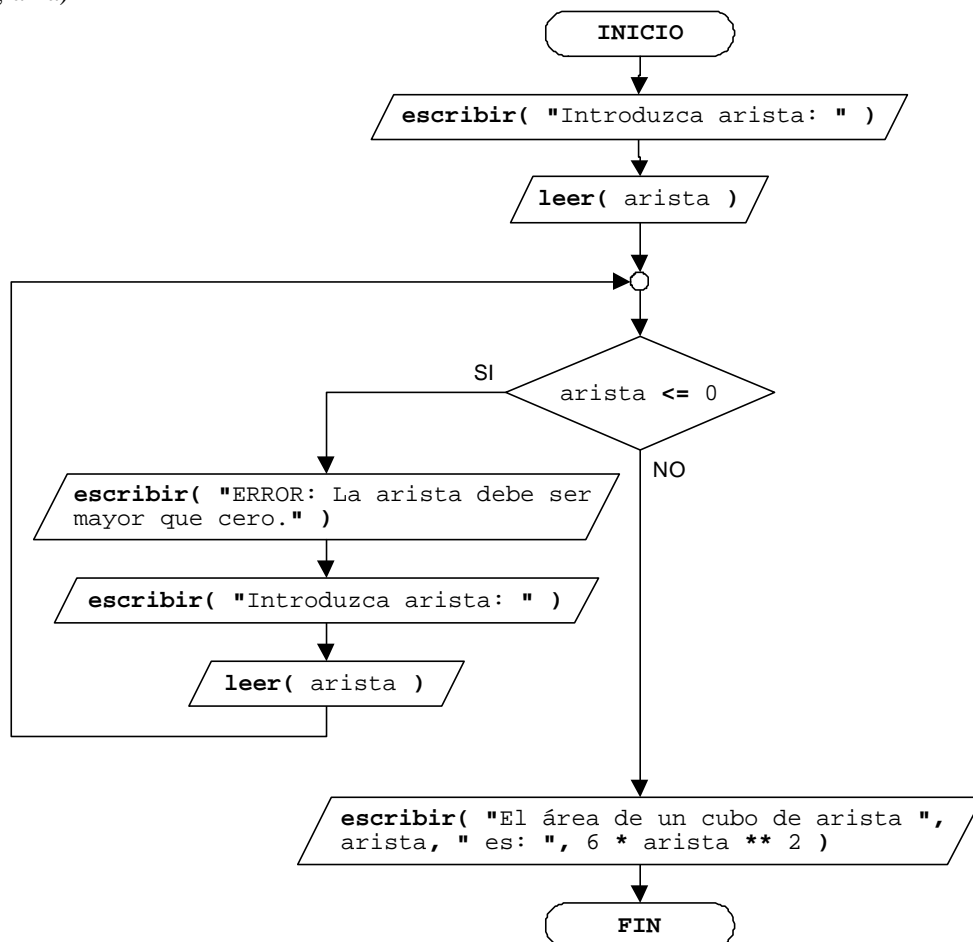
(Pseudocódigo)

```

Algoritmo Area_de_un_cubo ( mientras ) ( ejer_pro_13_01.c )
1  algoritmo Area_de_un_cubo
2
3  variables
4    real arista
5
6  inicio
7    escribir( "Introduzca arista: " )
8    leer( arista )
9
10   /* Filtramos la arista */
11
12   mientras ( arista <= 0 )
13     escribir( "ERROR: La arista debe ser mayor que cero." )
14     escribir( "Introduzca arista: " )
15     leer( arista )
16   fin_mientras
17
18   escribir( "El área de un cubo de arista ", arista, " es: ",
19           6 * arista ** 2 )
20  fin

```

(Ordinograma)



EJERCICIO PROPUESTO 13.2

Áreas de triángulos

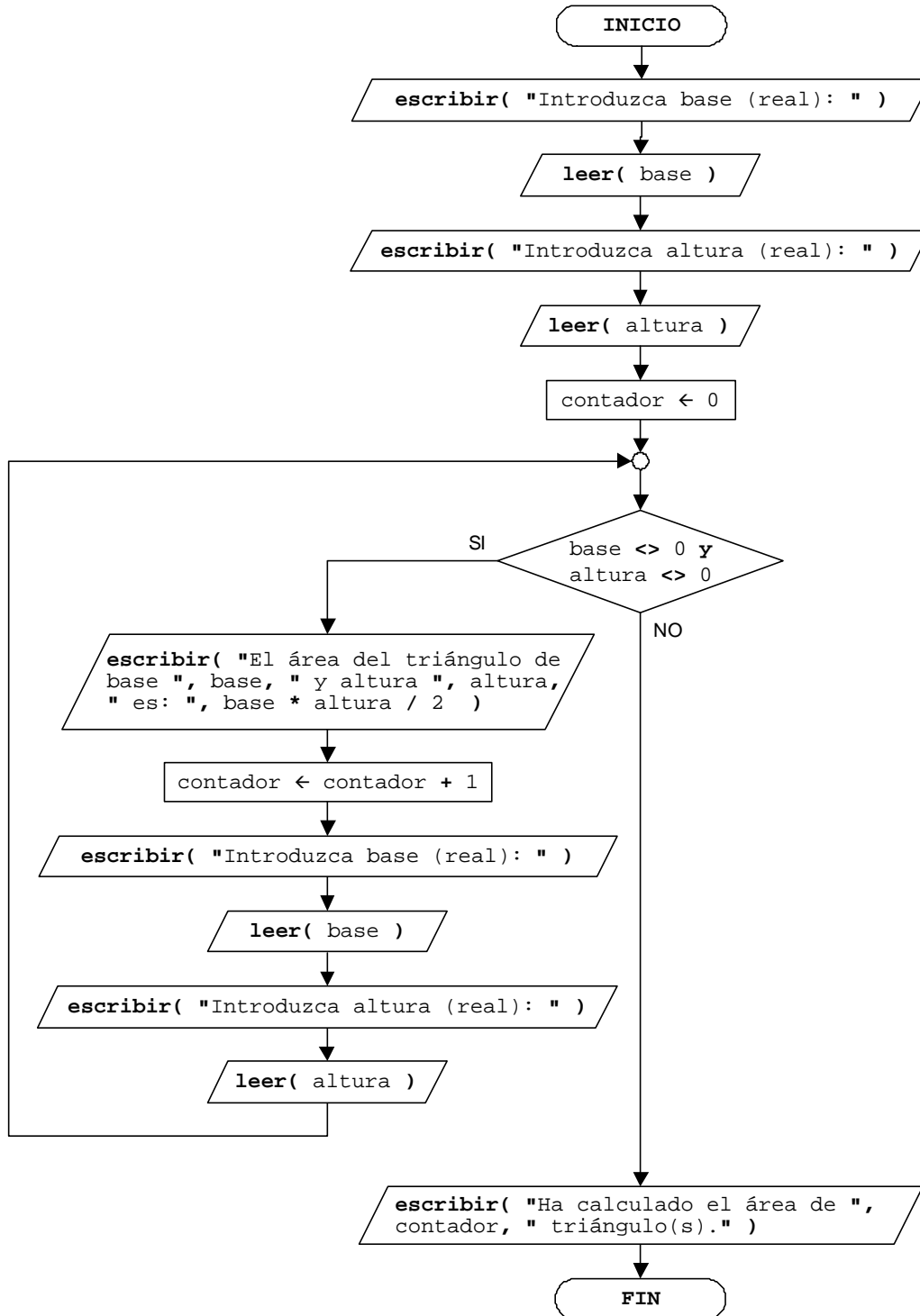
Solución:

(Pseudocódigo)

Algoritmo Areas_de_triangulos (mientras) (ejer_pro_13_02.c)	
1	algoritmo Areas_de_triangulos
2	
3	variables
4	entero contador
5	real altura, base
6	
7	inicio
8	escribir ("Introduzca base (real): ")
9	leer (base)
10	escribir ("Introduzca altura (real): ")
11	leer (altura)
12	contador ← 0
13	mientras (base <> 0 y altura <> 0)
14	escribir ("El área del triángulo de base ", base, " y altura ",
15	altura, " es: ", base * altura / 2)
16	contador ← contador + 1
17	escribir ("Introduzca base: ")
18	leer (base)
19	escribir ("Introduzca altura: ")
20	leer (altura)
21	fin_mientras
22	escribir ("Ha calculado el área de ", contador, " triángulo(s).")
23	fin

4 Empezar de cero a programar en lenguaje C

(Ordinograma)



EJERCICIO PROPUESTO 13.3

Traza y salida por pantalla (bucle **mientras**)

Solución:

La traza del algoritmo es:

<u>Secuencia:</u>	<u>Acción (instrucción):</u>	<u>Valor de:</u>	
		i	r
1	i = 5	5	?
2	(Comprobar si i >= 3)	5	?
La condición es verdadera . Inicio de la iteración 1.			
3	r = i + 3	5	8
4	escribir(r)	5	8
5	i = i - 1	4	8
Fin de la iteración 1.			
6	(Comprobar si i >= 3)	4	8
La condición es verdadera . Inicio de la iteración 2.			
7	r = i + 3	4	7
8	escribir(r)	4	7
9	i = i - 1	3	7
Fin de la iteración 2.			
10	(Comprobar si i >= 3)	3	7
La condición es verdadera . Inicio de la iteración 3.			
11	r = i + 3	3	6
12	escribir(r)	3	6
13	i = i - 1	2	6
Fin de la iteración 3.			
14	(Comprobar si i >= 3)	2	6
La condición es falsa . El bucle finaliza después de 3 iteraciones.			

Traza del ejercicio propuesto 13.3

6 Empezar de cero a programar en lenguaje C

La salida por pantalla es:

8 7 6

EJERCICIO PROPUESTO 13.4

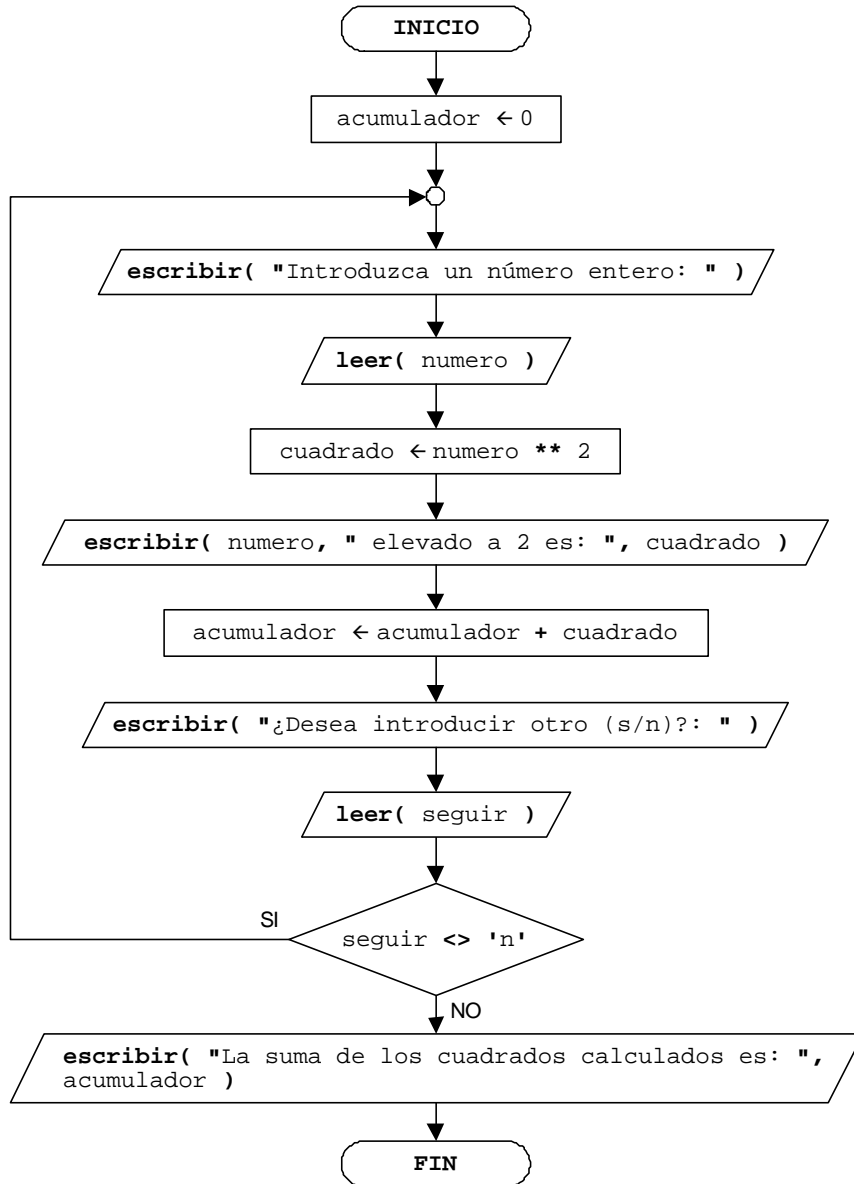
Suma de los cuadrados calculados

Solución:

(Pseudocódigo)

```
Algoritmo Suma_de_los_cuadrados_calculados (hacer...mientras) (ejer_pro_13_04.c)  
1 algoritmo Suma_de_los_cuadrados_calculados  
2  
3 variables  
4     caracter seguir  
5     entero acumulador, cuadrado, numero  
6  
7 inicio  
8     acumulador ← 0  
9     hacer  
10        escribir( "Introduzca un número entero: " )  
11        leer( numero )  
12        cuadrado ← numero ** 2  
13        escribir( numero, " elevado a 2 es: ", cuadrado )  
14        acumulador ← acumulador + cuadrado  
15        escribir( "¿Desea introducir otro (s/n)?: " )  
16        leer( seguir )  
17        mientras ( seguir <> 'n' )  
18        escribir( "La suma de los cuadrados calculados es: ", acumulador )  
19 fin
```

(Ordinograma)



EJERCICIO PROPUESTO 13.5

Suma y cantidad de resultados calculados

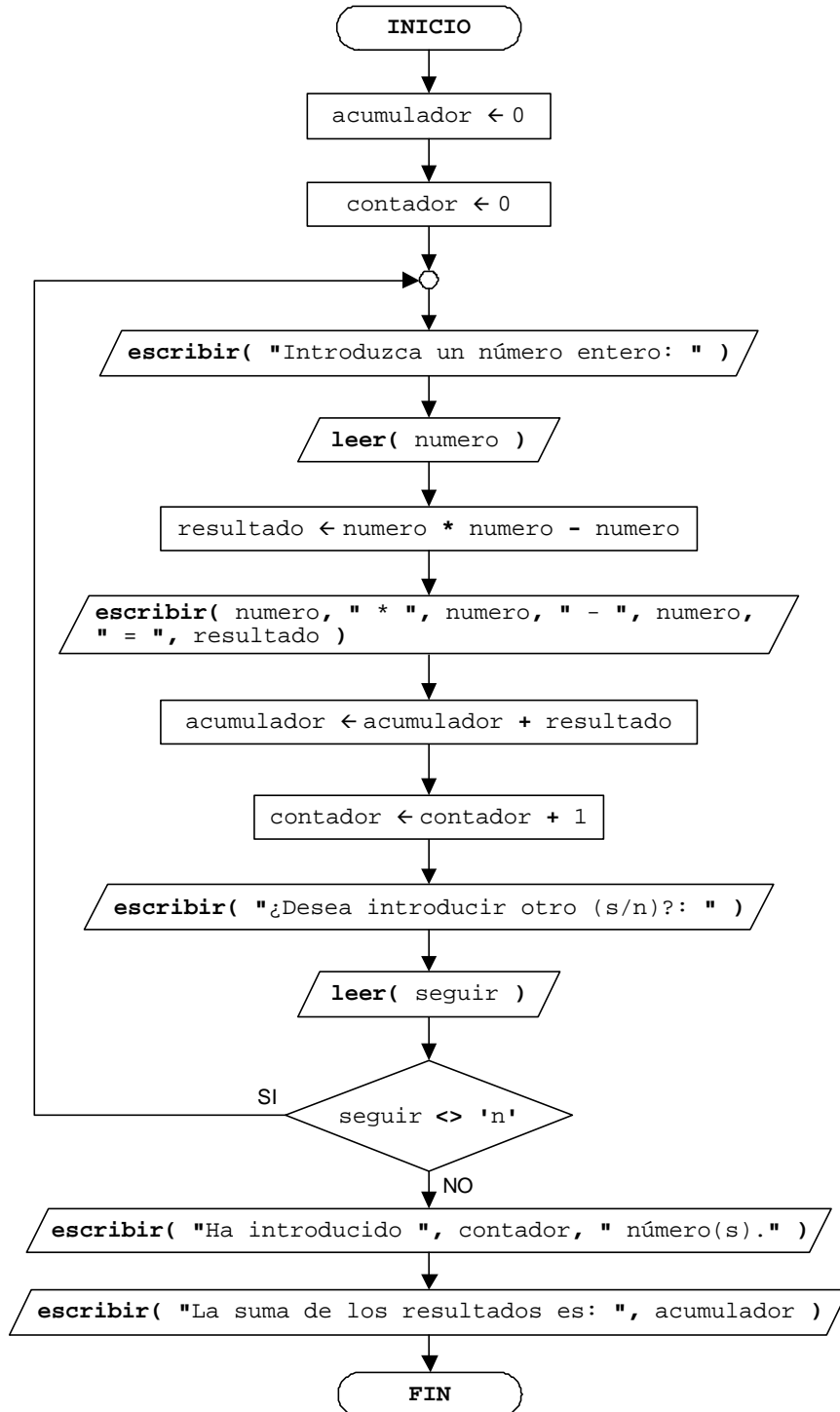
Solución:

(Pseudocódigo)

```

Algoritmo Suma_y_cantidad_de_resultados_calculados
( hacer...mientras ) (ejer_pro_13_05.c)
1 algoritmo Suma_y_cantidad_de_resultados_calculados
2
3 variables
4   caracter seguir
5   entero acumulador, contador, numero, resultado
6
7 inicio
8   acumulador ← 0
9   contador ← 0
10  hacer
11    escribir( "Introduzca un número entero: " )
12    leer( numero )
13
14    resultado ← numero * numero - numero
15    escribir( numero, " * ", numero, " - ", numero, " = ", resultado )
16
17    acumulador ← acumulador + resultado
18    contador ← contador + 1
19
20    escribir( "¿Desea introducir otro (s/n)?: " )
21    leer( seguir )
22  mientras ( seguir <> 'n' )
23    escribir( "Ha introducido ", contador, " número(s)." )
24    escribir( "La suma de los resultados es: ", acumulador )
25 fin
```

(Ordinograma)



EJERCICIO PROPUESTO 13.6

Traza y salida por pantalla (bucle `hacer...mientras`)**Solución:**

La traza del algoritmo es:

<u>Secuencia:</u>	<u>Acción (instrucción):</u>	<u>Valor de:</u>	
		i	r
1	<code>i = -5</code>	-5	?
2	<code>r = 10</code>	-5	10
Inicio de la iteración 1.			
3	<code>r = i * 10 - r</code>	-5	-60
4	<code>escribir(r)</code>	-5	-60
5	<code>i = i + 2</code>	-3	-60
Fin de la iteración 1.			
6	(Comprobar si <code>i <= -1</code>)	-3	-60
La condición es verdadera . Inicio de la iteración 2.			
7	<code>r = i * 10 - r</code>	-3	30
8	<code>escribir(r)</code>	-3	30
9	<code>i = i + 2</code>	-1	30
Fin de la iteración 2.			
10	(Comprobar si <code>i <= -1</code>)	-1	30
La condición es verdadera . Inicio de la iteración 3.			
11	<code>r = i * 10 - r</code>	-1	-40
12	<code>escribir(r)</code>	-1	-40
13	<code>i = i + 2</code>	1	-40
Fin de la iteración 3.			
14	(Comprobar si <code>i <= -1</code>)	1	-40
La condición es falsa . El bucle finaliza después de 3 iteraciones.			

Traza del ejercicio propuesto 13.6

La salida por pantalla es:

```
- 60 30 - 40
```

EJERCICIO PROPUESTO 13.7

Sucesión de números 1 -1 2 -2 3 -3 4 -4 5 -5

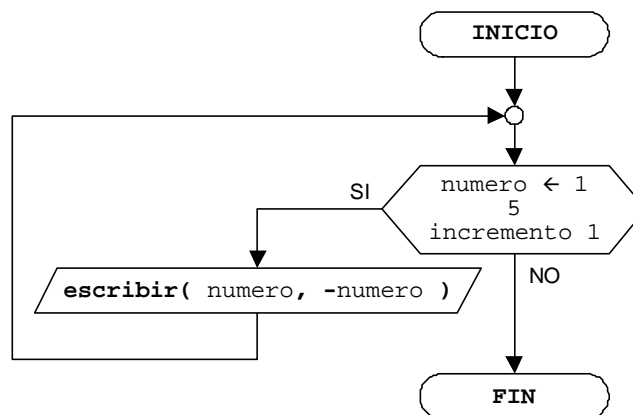
Solución:

(Pseudocódigo)

```

Algoritmo Sucesion_de_numeros ( para ) (ejer_pro_13_07.c)
1  algoritmo Sucesion_de_numeros
2
3  variables
4      entero numero
5
6  inicio
7      para numero ← 1 hasta 5 hacer
8          escribir( numero, -numero )
9      fin_para
10 fin
    
```

(Ordinograma)



EJERCICIO PROPUESTO 13.8

Primeras diez potencias de 2

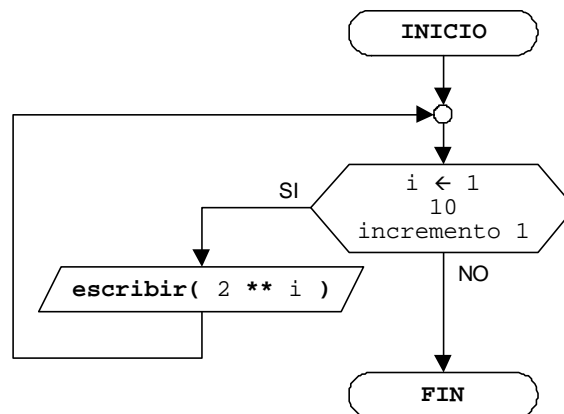
Solución:

(Pseudocódigo)

```

Algoritmo Primeras_diez_potencias_de_2 ( para ) (ejer_pro_13_08.c)
1  algoritmo Primeras_diez_potencias_de_2
2
3  variables
4      entero i
5
6  inicio
7      para i = 1 hasta 10 hacer
8          escribir( 2 ** i )
9      fin_para
10 fin
    
```

(Ordinograma)



EJERCICIO PROPUESTO 13.9

Números impares del -9 al 3

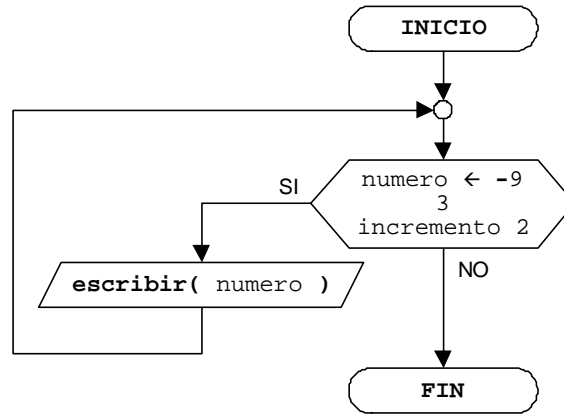
Solución:

(Pseudocódigo)

```

Algoritmo Numeros_impares_del_9_al_3 ( para ) (ejer_pro_13_09.c)
1  algoritmo Numeros_impares_del_menos_9_al_3
2
3  variables
4      entero numero
5
6  inicio
7      para numero = -9 hasta 3 incremento 2 hacer
8          escribir( numero )
9      fin_para
10 fin
    
```

(Ordinograma)



EJERCICIO PROPUESTO 13.10

Números pares del -50 al -80

Solución:

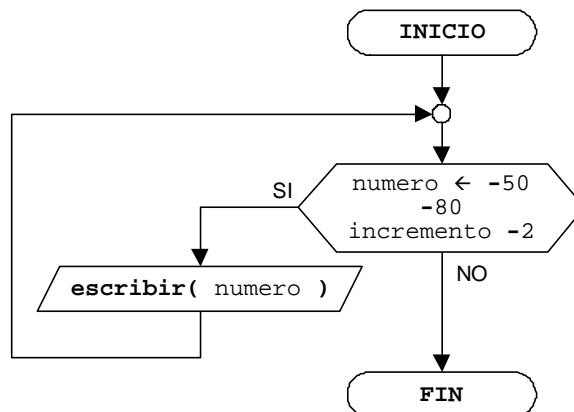
(Pseudocódigo)

Algoritmo Numeros_pares_del_menos_50_al_menos_80 (para) (ejer_pro_13_10.c)

```

1  algoritmo Numeros_pares_del_menos_50_al_menos_80
2
3  variables
4    entero numero
5
6  inicio
7    para numero = -50 hasta -80 incremento -2 hacer
8      /* También es correcto escribir: decremento 2 */
9      escribir( numero )
10   fin_para
11  fin
  
```

(Ordinograma)



EJERCICIO PROPUESTO 13.11

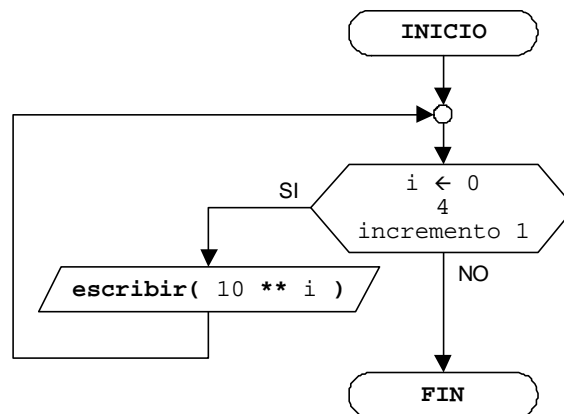
Sucesión de números 1 10 100 1000 10000

Solución:

(Pseudocódigo)

```
Algoritmo Sucesion_de_numeros ( para ) (ejer_pro_13_11.c)
1 algoritmo Sucesion_de_numeros
2
3 variables
4     entero i
5
6 inicio
7     para i ← 0 hasta 4 hacer
8         escribir( 10 ** i )
9     fin_para
10 fin
```

(Ordinograma)



EJERCICIO PROPUESTO 13.12

Traza y salida por pantalla (bucle para)

Solución:

La traza del algoritmo es:

<u>Secuencia:</u>	<u>Acción (instrucción):</u>	<u>Valor de:</u>	
		i	r
1	$r = 8$?	8
2	escribir(r)	?	8
3	$i = -1$	-1	8
4	(Comprobar si i es mayor o igual que -3)	-1	8
	i sí es mayor o igual que -3. Inicio de la iteración 1.		
5	$r = r - i$	-1	9
6	escribir(r)	-1	9
	Fin de la iteración 1.		
7	(Sumar a i el valor -3) (Restarle 1)	-2	9
8	(Comprobar si i es mayor o igual que -3)	-2	9
	i sí es mayor o igual que -3. Inicio de la iteración 2.		
9	$r = r - i$	-2	11
10	escribir(r)	-2	11
	Fin de la iteración 2.		
11	(Sumar a i el valor -1) (Restarle 1)	-3	11
12	(Comprobar si i es mayor o igual que -3)	-3	11
	i sí es mayor o igual que -3. Inicio de la iteración 3.		
13	$r = r - i$	-3	14
14	escribir(r)	-3	14
	Fin de la iteración 3.		
15	(Sumar a i el valor -1) (Restarle 1)	-4	14
16	(Comprobar si i es mayor o igual que -3)	-4	14
	i no es mayor o igual que -3. El bucle finaliza después de 3 iteraciones.		

Traza del ejercicio propuesto 13.12

16 Empezar de cero a programar en lenguaje C

La salida por pantalla es:

```
8 9 11 14
```

EJERCICIO PROPUESTO 13.13

Suma de números pares e impares

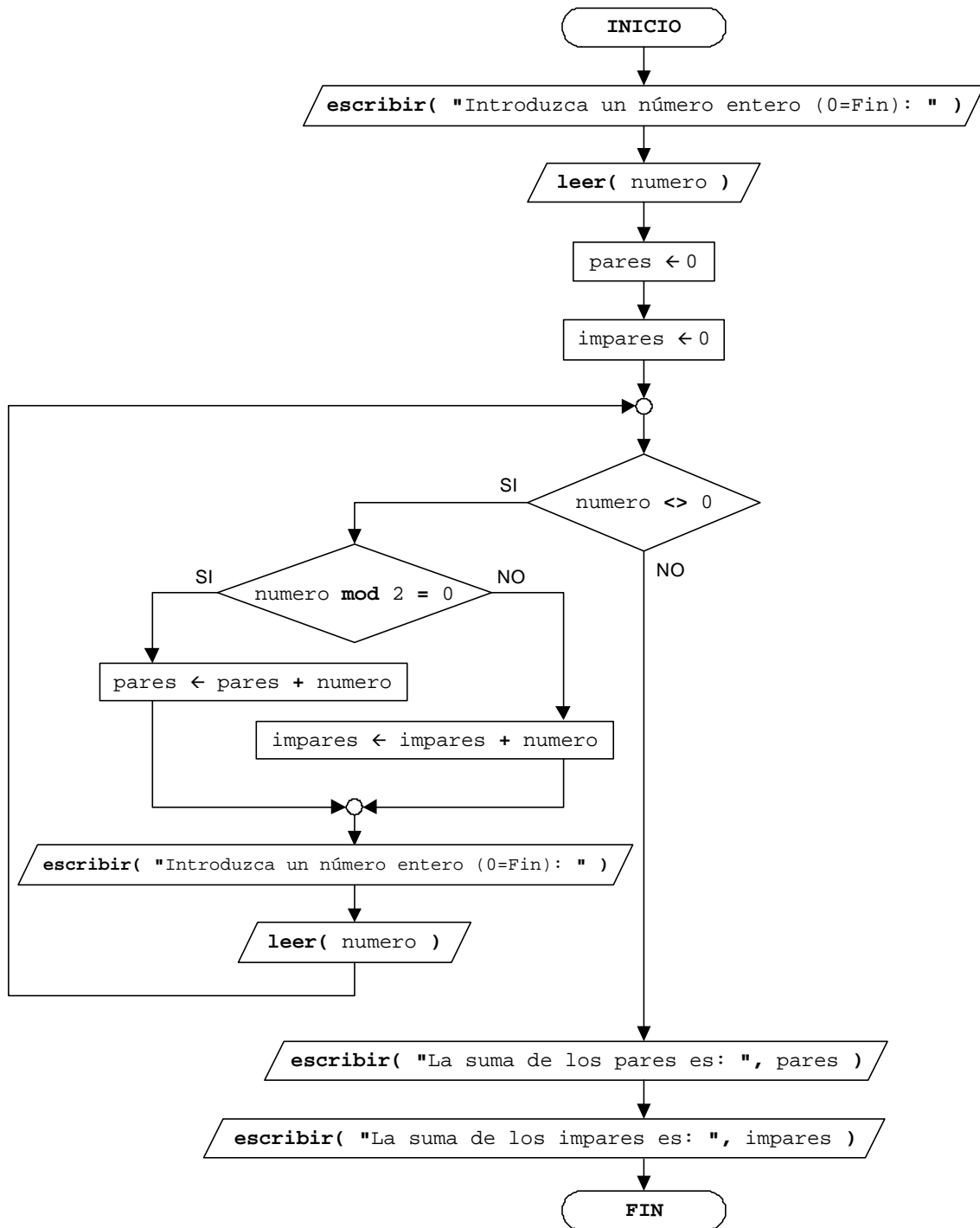
Solución:

(Pseudocódigo)

```

Algoritmo Suma_de_numeros_pares_e_impares (ejer_pro_13_13.c)
1  algoritmo Suma_de_numeros_pares_e_impares
2
3  variables
4      entero numero, pares, impares
5
6  inicio
7      escribir( "Introduzca un número entero (0=Fin): " )
8      leer( numero )
9
10     pares ← 0
11     impares ← 0
12
13     mientras ( numero <> 0 )
14
15         si ( numero mod 2 = 0 )
16             pares ← pares + numero
17         sino
18             impares ← impares + numero
19         fin_si
20
21     escribir( "Introduzca un número entero (0=Fin): " )
22     leer( numero )
23
24     fin_mientras
25
26     escribir( "La suma de los pares es: ", pares )
27     escribir( "La suma de los impares es: ", impares )
28 fin
```

(Ordinograma)



EJERCICIO PROPUESTO 13.14

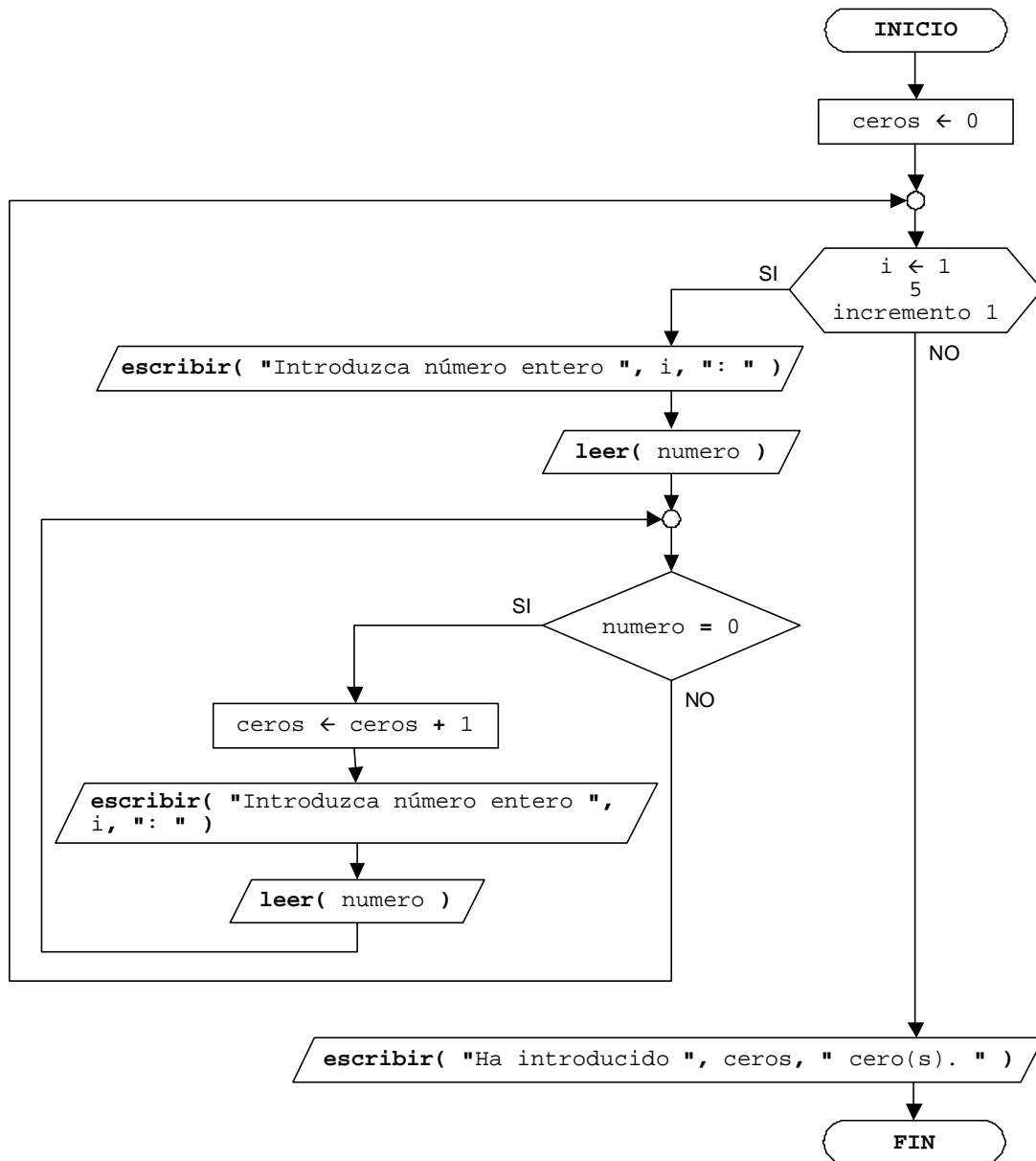
Ceros introducidos

Solución:

(Pseudocódigo)

Algoritmo Ceros_introducidos (Solución 1) (ejer_pro_13_14_1.c)	
1	algoritmo Ceros_introducidos
2	
3	variables
4	entero ceros, i, numero
5	
6	inicio
7	ceros ← 0
8	
9	para i ← 1 hasta 5 hacer
10	escribir ("Introduzca número entero", i, ": ")
11	leer (numero)
12	
13	mientras (numero = 0)
14	ceros ← ceros + 1
15	escribir ("Introduzca número entero", i, ": ")
16	leer (numero)
17	fin_mientras
18	fin_para
19	
20	escribir ("Ha introducido ", ceros, " cero(s).")
21	fin

(Ordinograma)



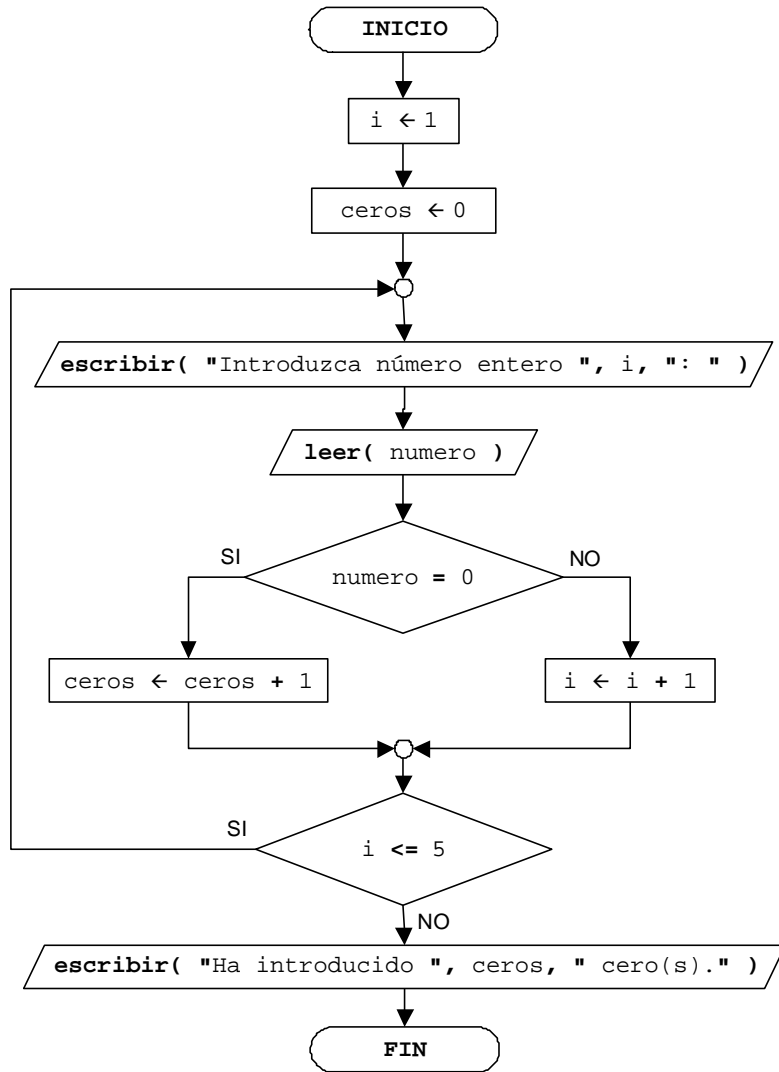
20 Empezar de cero a programar en lenguaje C

Una segunda **solución** es:

(Pseudocódigo)

Algoritmo Ceros_introducidos (Solución 2) (ejer_pro_13_14_2.c)	
1	algoritmo Ceros_introducidos
2	
3	variables
4	entero ceros, i, numero
5	
6	inicio
7	i ← 1
8	ceros ← 0
9	
10	hacer
11	escribir ("Introduzca número entero ", i, ": ")
12	leer (numero)
13	
14	si (numero = 0)
15	ceros ← ceros + 1
16	sino
17	i ← i + 1
18	fin_si
19	mientras (i <= 5)
20	
21	escribir ("Ha introducido ", ceros, " cero(s).")
22	fin

(Ordinograma)



EJERCICIO PROPUESTO 13.15

Número mayor introducido y su posición

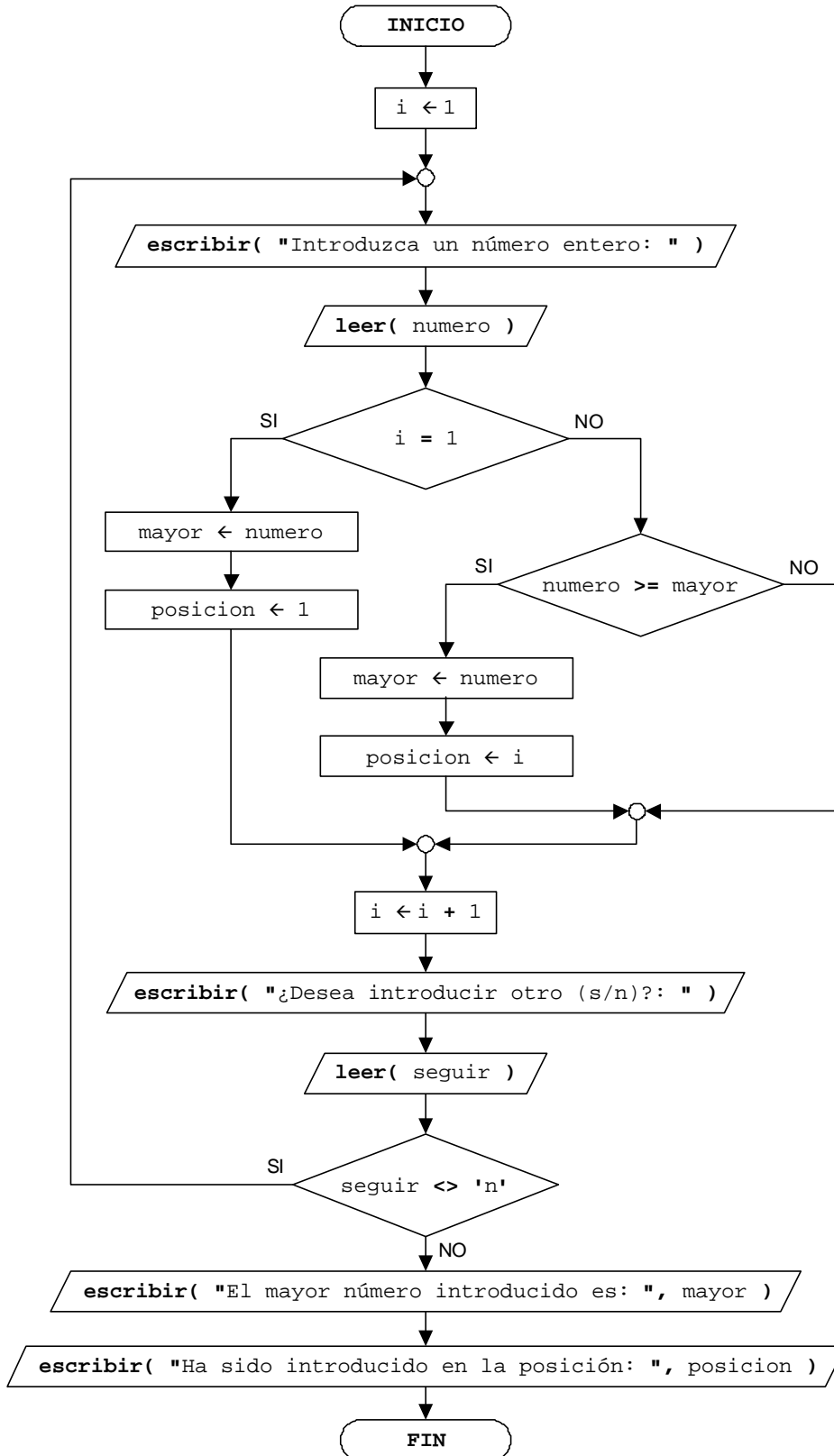
Solución:

(Pseudocódigo)

```

Algoritmo Numero_mayor_introducido_y_su_posicion (Solución 1) (ejer_pro_13_15_1.c)
1  algoritmo Numero_mayor_introducido_y_su_posicion
2
3  variables
4      caracter seguir
5      entero i, numero, mayor, posicion
6
7  inicio
8      i ← 1
9      hacer
10     escribir( "Introduzca un número entero: " )
11     leer( numero )
12
13     si ( i = 1 )
14         mayor ← numero
15         posicion ← 1
16     sino
17         si ( numero >= mayor )
18             mayor ← numero
19             posicion ← i
20         fin_si
21     fin_si
22
23     i ← i + 1
24
25     escribir( "¿Desea introducir otro (s/n)?:" )
26     leer( seguir )
27     mientras ( seguir <> 'n' )
28
29     escribir( "El mayor número introducido es: ", mayor )
30     escribir( "Ha sido introducido en la posicion: ", posicion )
31 fin
```

(Ordinograma)



24 Empezar de cero a programar en lenguaje C

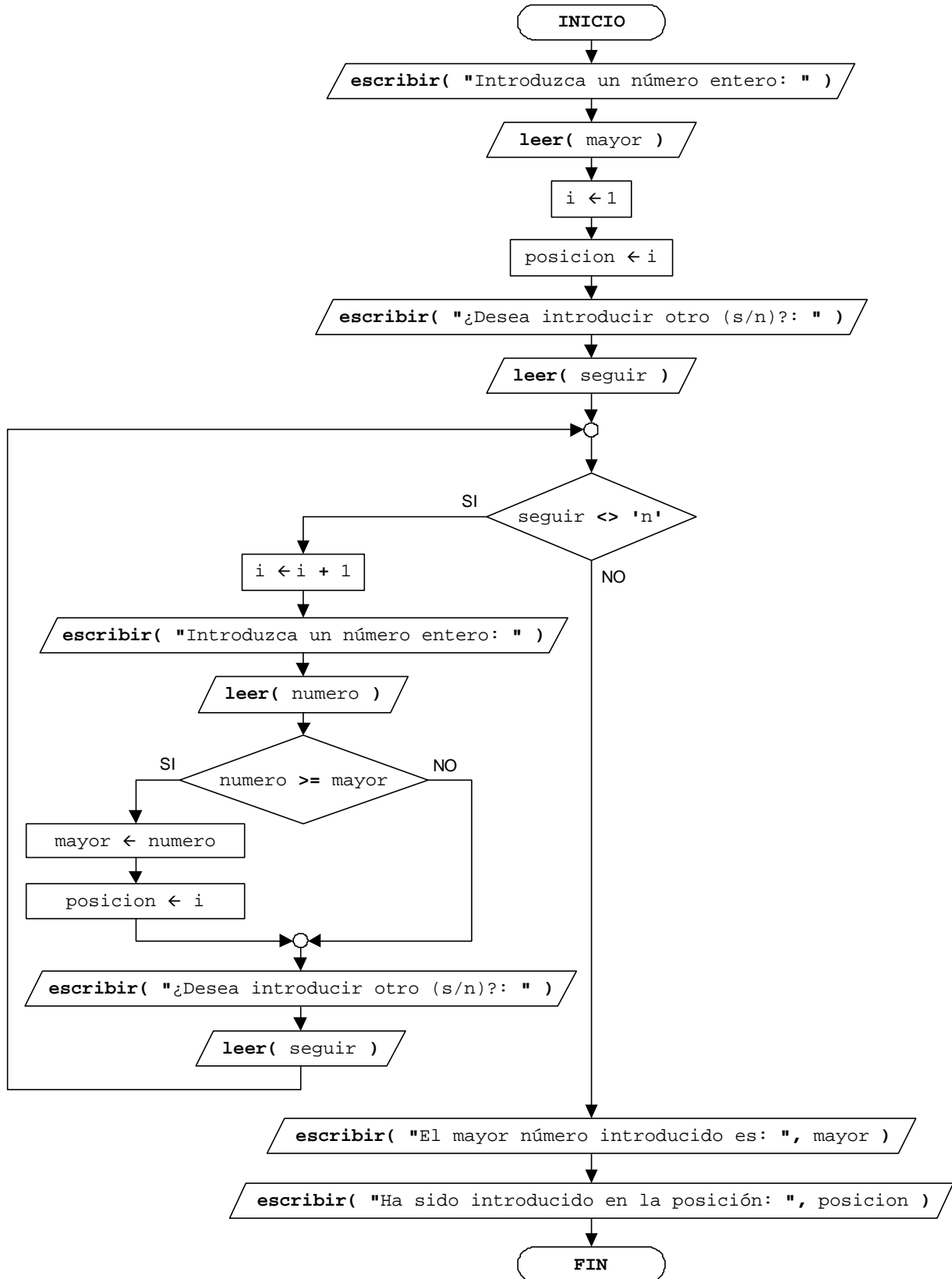
Una segunda **solución** es:

(Pseudocódigo)

```

Algoritmo Numero_mayor_introducido_y_su_posicion (Solución 2) ( ejer_pro_13_15_2.c)
1  algoritmo Numero_mayor_introducido_y_su_posicion
2
3  variables
4      caracter seguir
5      entero i, numero, mayor, posicion
6
7  inicio
8      escribir( "Introduzca un número entero: " )
9      leer( mayor )
10
11     i ← 1
12     posicion ← i
13
14     escribir( "¿Desea introducir otro (s/n)?: " )
15     leer( seguir )
16
17     mientras ( seguir <> 'n' )
18         i ← i + 1
19
20         escribir( "Introduzca un número entero: " )
21         leer( numero )
22
23         si ( numero >= mayor )
24             mayor ← numero
25             posicion ← i
26         fin_si
27
28         escribir( "¿Desea introducir otro (s/n)?: " )
29         leer( seguir )
30     fin_mientras
31
32     escribir( "El mayor número introducido es: ", mayor )
33     escribir( "Ha sido introducido en la posicion: ", posicion )
34 fin
```

(Ordinograma)



EJERCICIO PROPUESTO 13.16

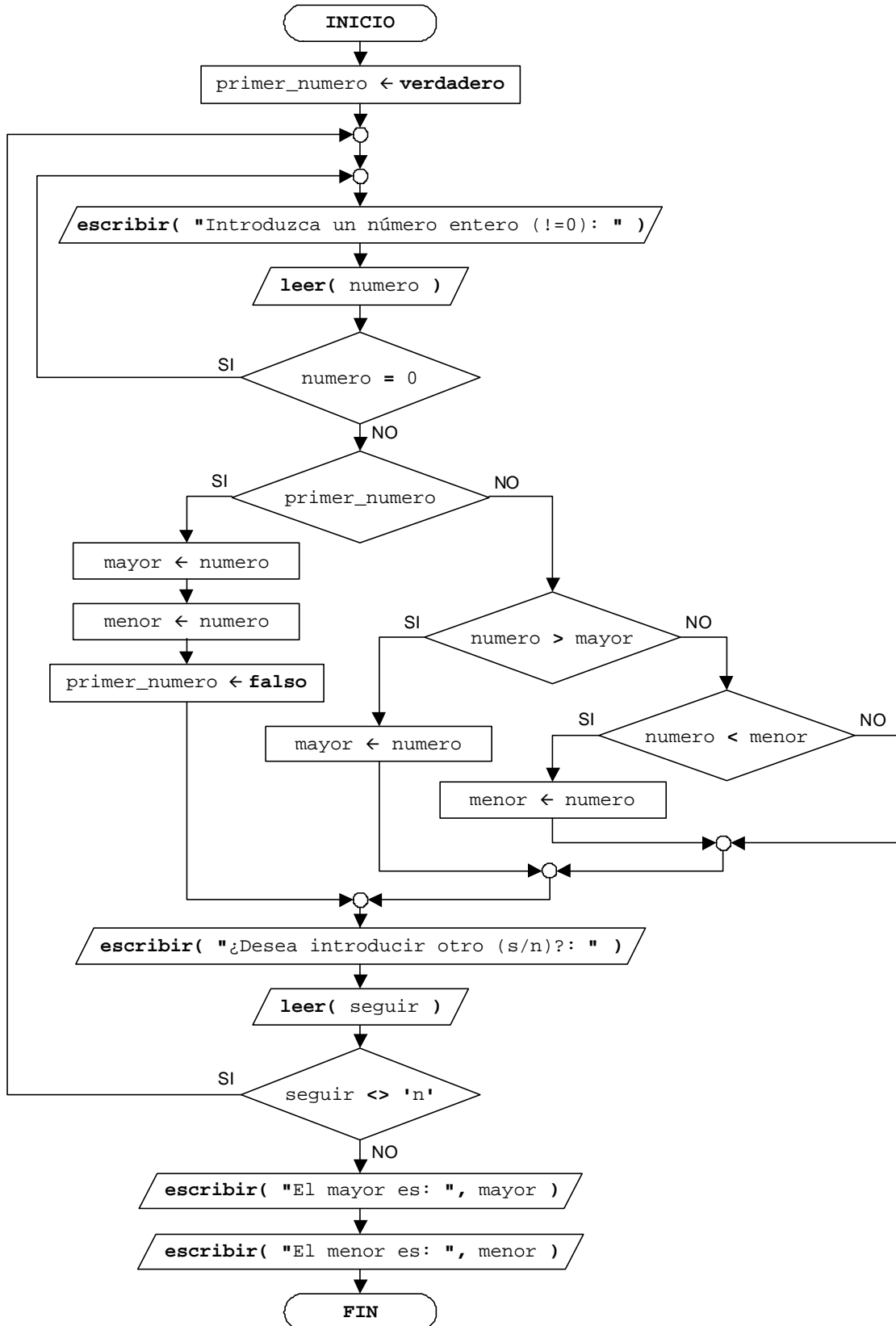
Número mayor y menor introducidos

Solución:

(Pseudocódigo)

Algoritmo Numero_mayor_y_menor_introducidos (Solución 1) (ejer_pro_13_16_1.c)	
1	algoritmo Numero_mayor_y_menor_introducidos
2	
3	variables
4	caracter seguir
5	entero numero, mayor, menor
6	logico primer_numero
7	
8	inicio
9	primer_numero \leftarrow verdadero
10	
11	hacer
12	
13	/* Filtramos el número */
14	hacer
15	escribir ("Introduzca un número entero (!=0): ")
16	leer (numero)
17	mientras (numero = 0)
18	/* Fin del filtro */
19	
20	si (primer_numero)
21	mayor \leftarrow numero
22	menor \leftarrow numero
23	primer_numero \leftarrow falso
24	sino
25	si (numero > mayor)
26	mayor \leftarrow numero
27	sino
28	si (numero < menor)
29	menor \leftarrow numero
30	fin_si
31	fin_si
32	fin_si
33	
34	escribir ("¿Desea introducir otro (s/n)?: ")
35	leer (seguir)
36	mientras (seguir \neq 'n')
37	
38	escribir ("El mayor es: ", mayor)
39	escribir ("El menor es: ", menor)
40	fin

(Ordinograma)



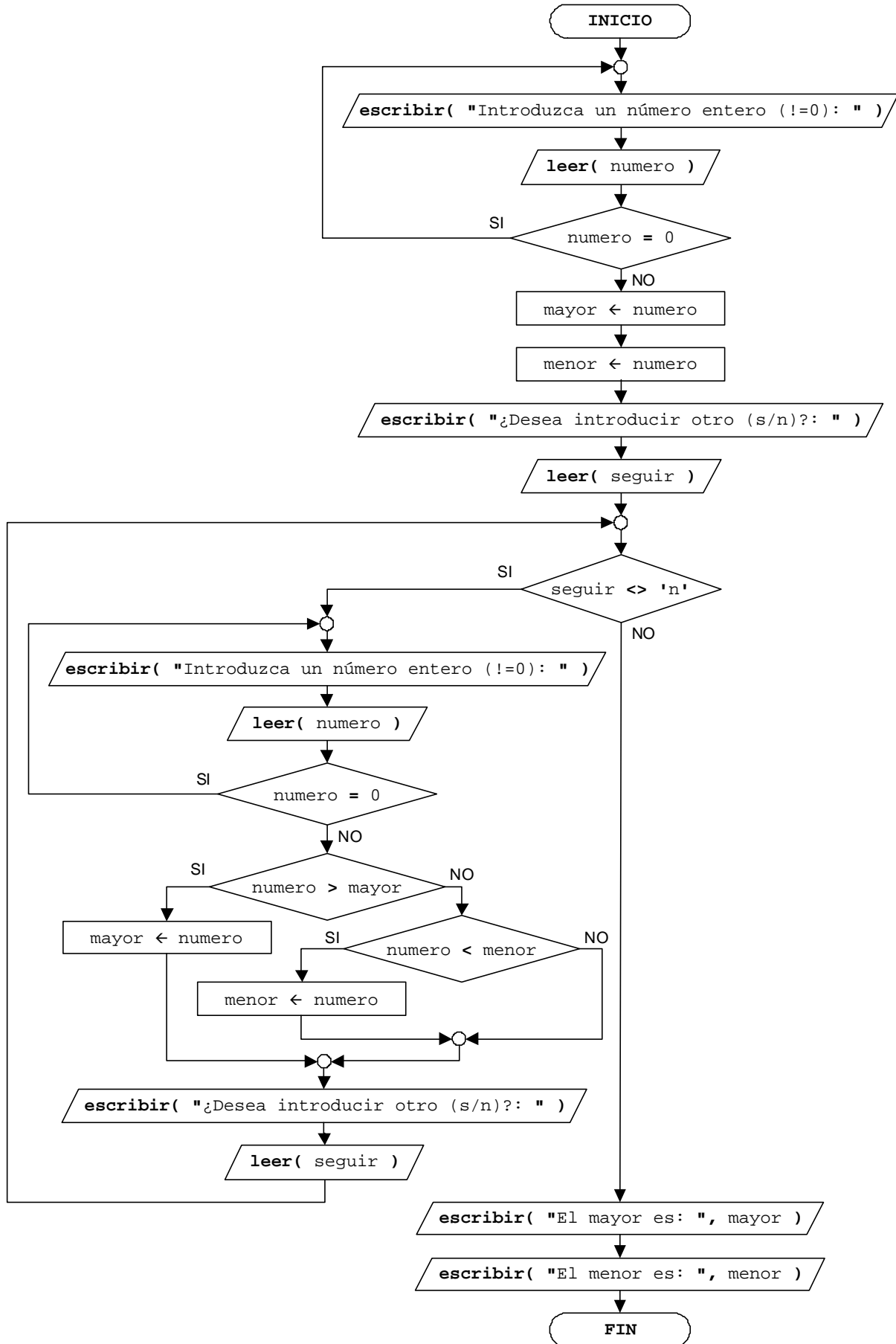
28 Empezar de cero a programar en lenguaje C

Una segunda **solución** es:

(Pseudocódigo)

```
Algoritmo Numero_mayor_y_menor_introducidos (Solución 2) (ejer_pro_13_16_2.c)
1 algoritmo Numero_mayor_y_menor_introducidos
2
3 variables
4     caracter seguir
5     entero numero, mayor, menor
6
7 inicio
8     hacer
9         escribir( "Introduzca un número entero (!=0): " )
10        leer( numero )
11    mientras ( numero = 0 )
12
13        mayor ← numero
14        menor ← numero
15
16        escribir( "¿Desea introducir otro (s/n)?: " )
17        leer( seguir )
18
19    mientras ( seguir <> 'n' )
20
21        hacer
22            escribir( "Introduzca un número entero (!=0): " )
23            leer( numero )
24            mientras ( numero = 0 )
25
26            si ( numero > mayor )
27                mayor ← numero
28            sino
29                si ( numero < menor )
30                    menor ← numero
31            fin_si
32        fin_si
33
34        escribir( "¿Desea introducir otro (s/n)?: " )
35        leer( seguir )
36    fin_mientras
37
38    escribir( "El mayor es: ", mayor )
39    escribir( "El menor es: ", menor )
40 fin
```

(Ordinograma)



EJERCICIO PROPUESTO 13.17

Traza y salida por pantalla (**hacer...mientras en mientras**)**Solución:**

La traza del algoritmo es:

<u>Secuencia:</u>	<u>Acción (instrucción):</u>	<u>Valor de:</u>		
		i	j	r
1	i = 3	3	?	?
2	(Comprobar si i <= 5)	3	?	?
	La condición es verdadera . Inicio de la iteración 1 del bucle 1.			
3	j = -2	3	-2	?
	Inicio de la iteración 1 del bucle 2.			
4	r = j - 1	3	-2	-5
5	escribir(r)	3	-2	-5
6	j = j - 1	3	-3	-5
	Fin de la iteración 1 del bucle 2.	3	-3	-5
7	(Comprobar si j >= -3)	3	-3	-5
	La condición es verdadera . Inicio de la iteración 2 del bucle 2.			
8	r = j - 1	3	-3	-6
9	escribir(r)	3	-3	-6
10	j = j - 1	3	-4	-6
	Fin de la iteración 2 del bucle 2.			
11	(Comprobar si j >= -3)	3	-4	-6
	La condición es falsa . El bucle 2 (anidado) finaliza después de 2 iteraciones.			
12	i = i + 2	5	-4	-6
	Fin de la iteración 1 del bucle 1.			
13	(Comprobar si i <= 5)	5	-4	-6
	La condición es verdadera . Inicio de la iteración 2 del bucle 1.			
14	j = -2	5	-2	-6

15	(Comprobar si $i \leq 5$)	5	-2	-6
	La condición es verdadera .			
	Inicio de la iteración 1 del bucle 2 (por segunda vez).			
16	$r = j - 1$	5	-2	-7
17	escribir (r)	5	-2	-7
18	$j = j - 1$	5	-3	-7
	Fin de la iteración 1 del bucle 2 (por segunda vez).			
19	(Comprobar si $j \leq -3$)	5	-3	-7
	La condición es verdadera .			
	Inicio de la iteración 2 del bucle 2 (por segunda vez).			
20	$r = j - 1$	5	-3	-8
21	escribir (r)	5	-3	-8
22	$j = j - 1$	5	-4	-8
	Fin de la iteración 2 del bucle 2 (por segunda vez).			
23	(Comprobar si $j \leq -3$)	5	-4	-8
	La condición es falsa .			
	El bucle 2 (anidado) vuelve a finalizar después de 2 iteraciones.			
24	$i = i + 2$	7	-4	-8
	Fin de la iteración 2 del bucle 1.			
25	(Comprobar si $i \leq 5$)	7	-4	-8
	La condición es falsa .			
	El bucle 1 finaliza después de 2 iteraciones.			

Traza del ejercicio resuelto 13.17

El bucle 1 itera dos (2) veces y, en cada una de ellas, el bucle 2 itera otras dos (2) veces, haciendo un total de cuatro iteraciones (2*2).

La salida por pantalla es:

- 5 - 6 - 7 - 8

EJERCICIO PROPUESTO 13.18

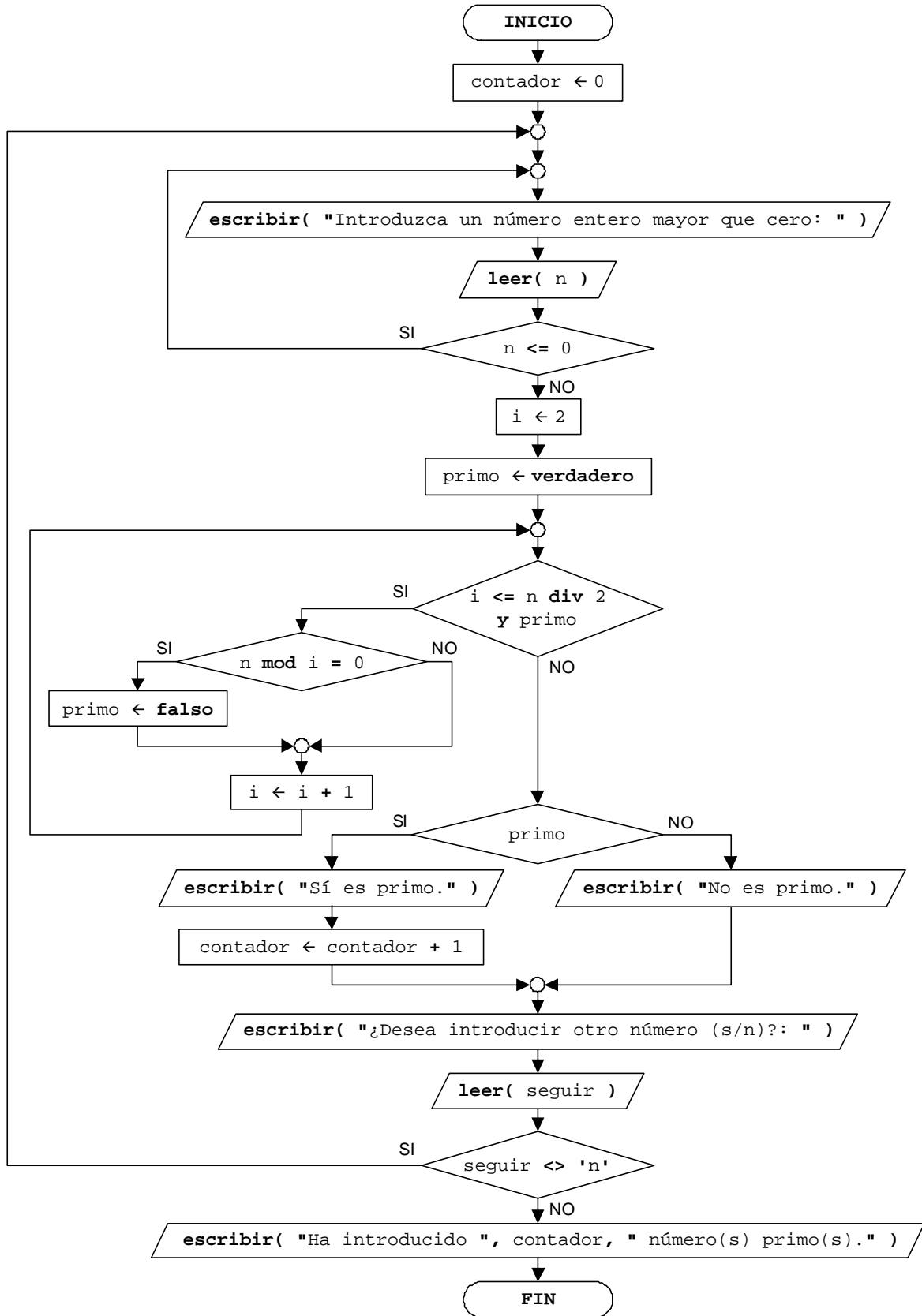
¿Cuántos números son primos?

Solución:

(Pseudocódigo)

Algoritmo Cuantos_numeros_son_primos (ejer_pro_13_18.c)	
1	algoritmo Cuantos_numeros_son_primos
2	
3	variables
4	caracter seguir
5	entero contador, i, n
6	logico primo
7	
8	inicio
9	contador ← 0
10	hacer
11	
12	/* Filtramos el número */
13	hacer
14	escribir ("Introduzca un número entero mayor que cero: ")
15	leer (n)
16	mientras (n ≤ 0)
17	
18	/* Comprobamos si el número es primo */
19	i ← 2
20	primo ← verdadero
21	
22	mientras (i ≤ n div 2 y primo)
23	si (n mod i = 0)
24	primo ← falso
25	fin_si
26	i ← i + 1
27	fin_mientras
28	
29	si (primo)
30	escribir ("Sí es primo.")
31	contador ← contador + 1
32	sino
33	escribir ("No es primo.")
34	fin_si
35	
36	escribir ("¿Desea introducir otro número (s/n)?:")
37	leer (seguir)
38	mientras (seguir <> 'n')
39	escribir ("Ha introducido ", contador, " número(s) primo(s).")
40	fin

(Ordinograma)



EJERCICIO PROPUESTO 13.19

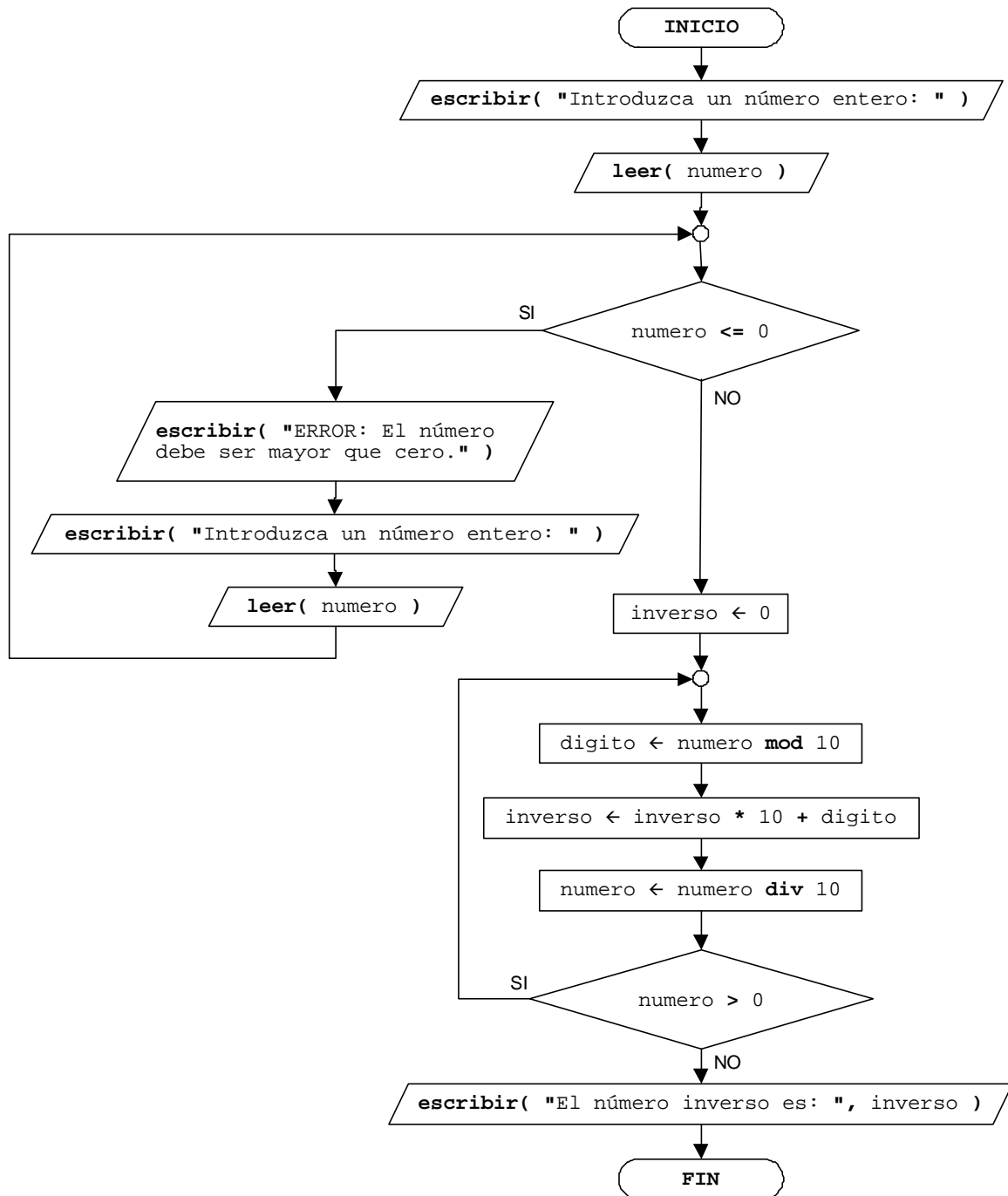
Número inverso

Solución:

(Pseudocódigo)

Algoritmo Numero_inverso (ejer_pro_13_19.c)	
1	algoritmo Numero_inverso
2	
3	variables
4	entero digito, inverso, numero
5	
6	inicio
7	escribir ("Introduzca un número entero: ")
8	leer (numero)
9	
10	/* Filtramos el número */
11	mientras (numero <= 0)
12	escribir ("ERROR: El número debe ser mayor que cero.")
13	escribir ("Introduzca un número entero: ")
14	leer (numero)
15	fin_mientras
16	
17	/* Calculamos el número inverso */
18	inverso ← 0
19	hacer
20	digito ← numero mod 10
21	inverso ← inverso * 10 + digito
22	numero ← numero div 10
23	mientras (numero > 0)
24	
25	escribir ("El número inverso es: ", inverso)
26	fin

(Ordinograma)



EJERCICIO PROPUESTO 13.20

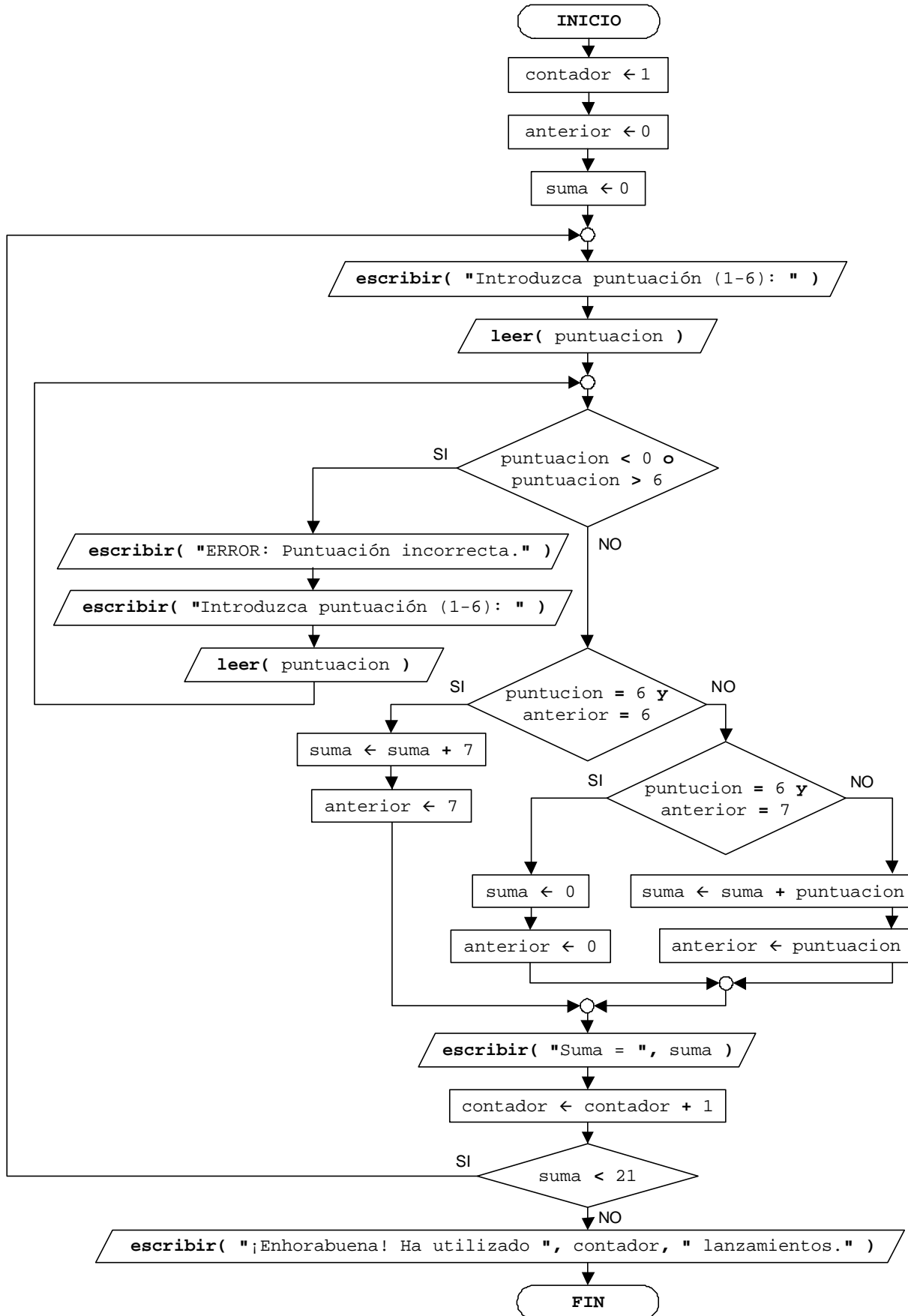
Lanzar un dado hasta sumar 21 ó más

Solución:

(Pseudocódigo)

Algoritmo Lanzar_un_dado_hasta_sumar_21_o_mas (ejer_pro_13_20.c)	
1	algoritmo Lanzar_un_dado_hasta_sumar_21_o_mas
2	
3	variables
4	entero anterior, contador, puntuacion, suma
5	
6	inicio
7	contador ← 1
8	anterior ← 0
9	suma ← 0
10	hacer
11	escribir ("Introduzca puntuación (1-6): ")
12	leer (puntuacion)
13	
14	/* Filtramos la puntuación */
15	mientras (puntuacion < 0 o puntuacion > 6)
16	escribir ("ERROR: Puntuación incorrecta.")
17	escribir ("Introduzca puntuación (1-6): ")
18	leer (puntuacion)
19	fin_mientras
20	
21	/* Actualizamos suma */
22	si (puntuacion = 6 y anterior = 6)
23	suma ← suma + 7
24	anterior ← 7
25	sino
26	si (puntuacion = 6 y anterior = 7)
27	suma ← 0
28	anterior ← 0
29	sino
30	suma ← suma + puntuacion
31	anterior ← puntuacion
32	fin_si
33	fin_si
34	
35	escribir ("Suma = ", suma)
36	contador ← contador + 1
37	mientras (suma < 21)
38	
39	escribir (";Enhorabuena! Ha utilizado ", contador,
40	" lanzamientos.")
41	fin

(Ordinograma)



EJERCICIO PROPUESTO 13.21

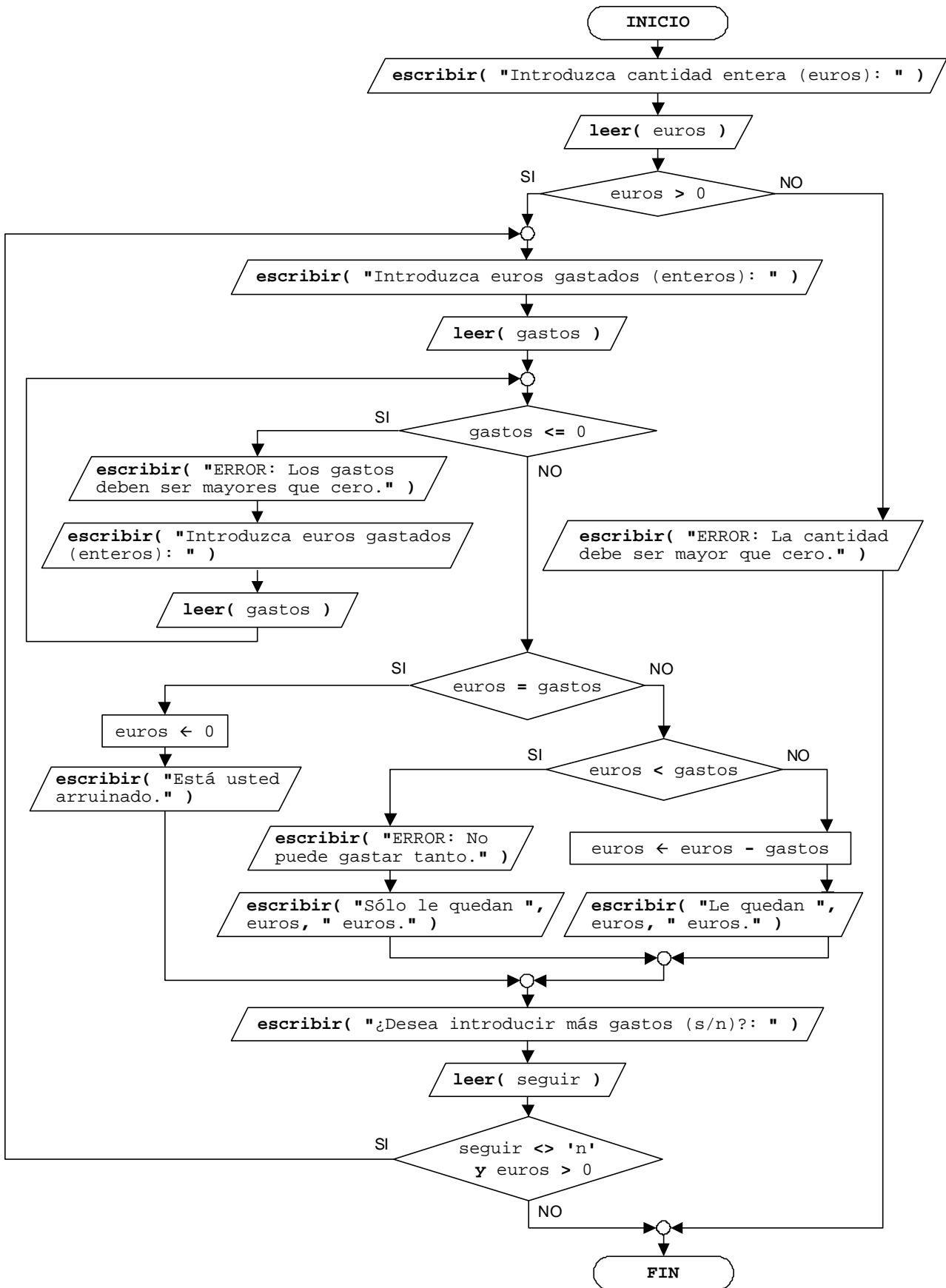
Gastos de un monedero

Solución:

(Pseudocódigo)

Algoritmo Gastos_de_un_monedero (ejer_pro_13_21.c)	
1	algoritmo Gastos_de_un_monedero
2	
3	variables
4	caracter seguir
5	entero euros, gastos
6	
7	inicio
8	escribir ("Introduzca cantidad entera (euros): ")
9	leer (euros)
10	si (euros > 0)
11	
12	hacer
13	escribir ("Introduzca euros gastados (enteros): ")
14	leer (gastos)
15	
16	/* Filtramos los gastos */
17	mientras (gastos <= 0)
18	escribir ("ERROR: Los gastos deben ser mayores que cero.")
19	escribir ("Introduzca euros gastados (enteros): ")
20	leer (gastos)
21	fin_mientras
22	
23	/* Calculamos la nueva cantidad disponible */
24	si (euros = gastos)
25	euros = 0
26	escribir ("Está usted arruinado.")
27	sino
28	
29	si (euros < gastos)
30	escribir ("ERROR: No puede gastar tanto.")
31	escribir ("Sólo le quedan ", euros, " euros.")
32	sino
33	euros = euros - gastos
34	escribir ("Le quedan ", euros, " euros.")
35	fin_si
36	
37	escribir ("¿Desea introducir más gastos (s/n)?: ")
38	leer (seguir)
39	
40	fin_si
41	
42	mientras (seguir <> 'n' y euros > 0)
43	
44	sino
45	escribir ("ERROR: La cantidad debe ser mayor que cero.")
46	fin_si
47	fin

(Ordinograma)



EJERCICIO PROPUESTO 13.22

Tiempo transcurrido entre dos horas

Solución:

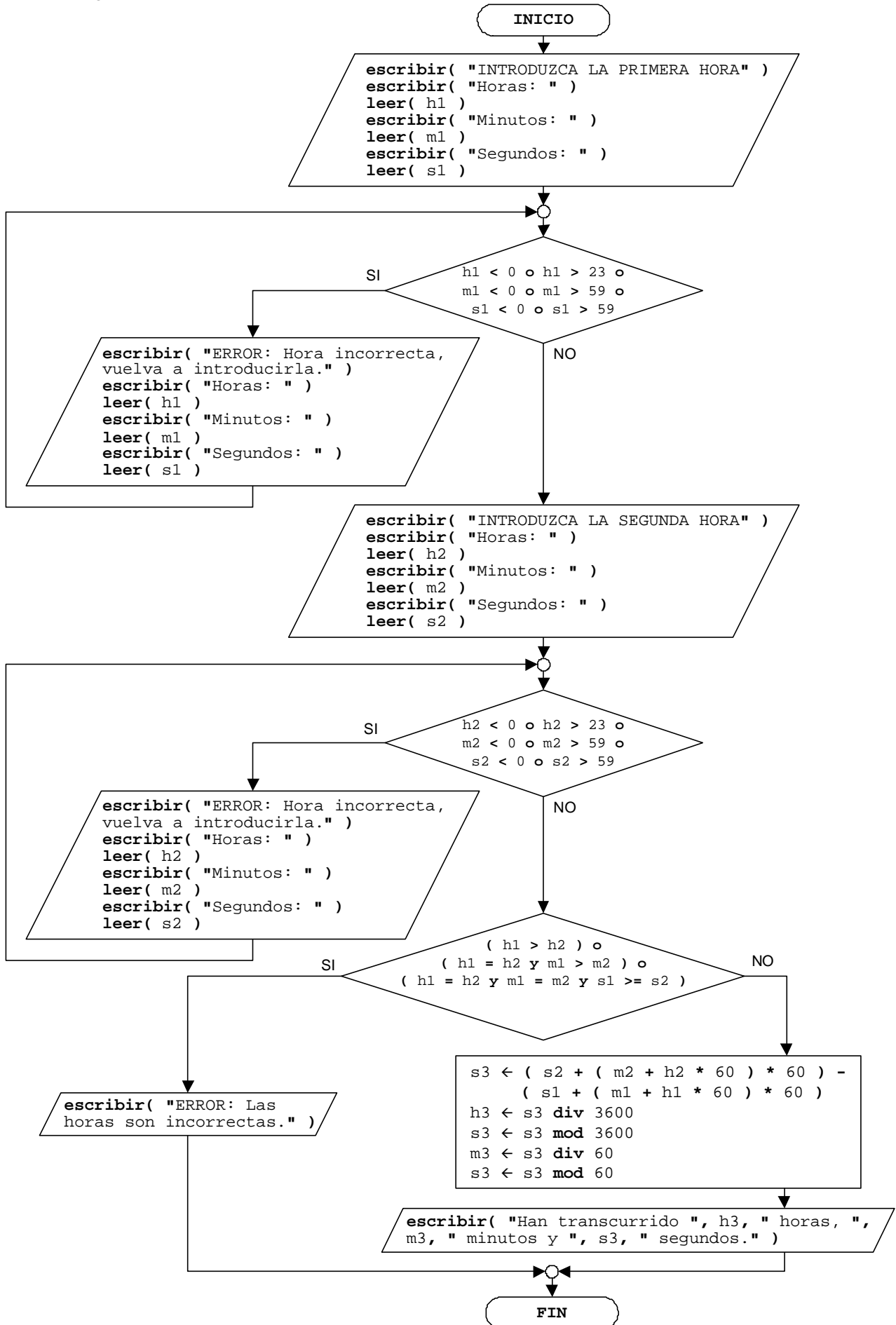
40 Empezar de cero a programar en lenguaje C

(Pseudocódigo)

```

Algoritmo Tiempo_transcurrido_entre_dos_horas (ejer_pro_13_22.c)
1  algoritmo Tiempo_transcurrido_entre_dos_horas
2
3  variables
4      entero h1, m1, s1, h2, m2, s2, h3, m3, s3
5
6  inicio
7      escribir( "INTRODUZCA LA PRIMERA HORA" )
8      escribir( "Horas: " )
9      leer( h1 )
10     escribir( "Minutos: " )
11     leer( m1 )
12     escribir( "Segundos: " )
13     leer( s1 )
14
15     /* Filtramos la primera hora */
16     mientras ( h1 < 0 o h1 > 23 o m1 < 0 o m1 > 59 o s1 < 0 o s1 > 59 )
17         escribir( "ERROR: Hora incorrecta, vuelva a introducirla." )
18         escribir( "Horas: " )
19         leer( h1 )
20         escribir( "Minutos: " )
21         leer( m1 )
22         escribir( "Segundos: " )
23         leer( s1 )
24     fin_mientras
25
26     escribir( "INTRODUZCA LA SEGUNDA HORA" )
27     escribir( "Horas: " )
28     leer( h2 )
29     escribir( "Minutos: " )
30     leer( m2 )
31     escribir( "Segundos: " )
32     leer( s2 )
33
34     /* Filtramos la segunda hora */
35     mientras ( h2 < 0 o h2 > 23 o m2 < 0 o m2 > 59 o s2 < 0 o s2 > 59 )
36         escribir( "ERROR: Hora incorrecta, vuelva a introducirla." )
37         escribir( "Horas: " )
38         leer( h2 )
39         escribir( "Minutos: " )
40         leer( m2 )
41         escribir( "Segundos: " )
42         leer( s2 )
43     fin_mientras
44
45     si ( ( h1 > h2 ) o
46         ( h1 = h2 y m1 > m2 ) o
47         ( h1 = h2 y m1 = m2 y s1 >= s2 ) )
48
49         escribir( "ERROR: Las horas son incorrectas." )
50     sino
51         s3 ← ( s2 + ( m2 + h2 * 60 ) * 60 ) - ( s1 + ( m1 + h1 * 60 ) * 60 )
52         h3 ← s3 div 3600
53         s3 ← s3 mod 3600
54         m3 ← s3 div 60
55         s3 ← s3 mod 60
56         escribir( "Han transcurrido ", h3, " horas, ", m3, " minutos y ",
57                 s3, " segundos." )
58     fin_si
59 fin
```

(Ordinograma)



EJERCICIO PROPUESTO 13.23

Desglose de moneda

Solución:

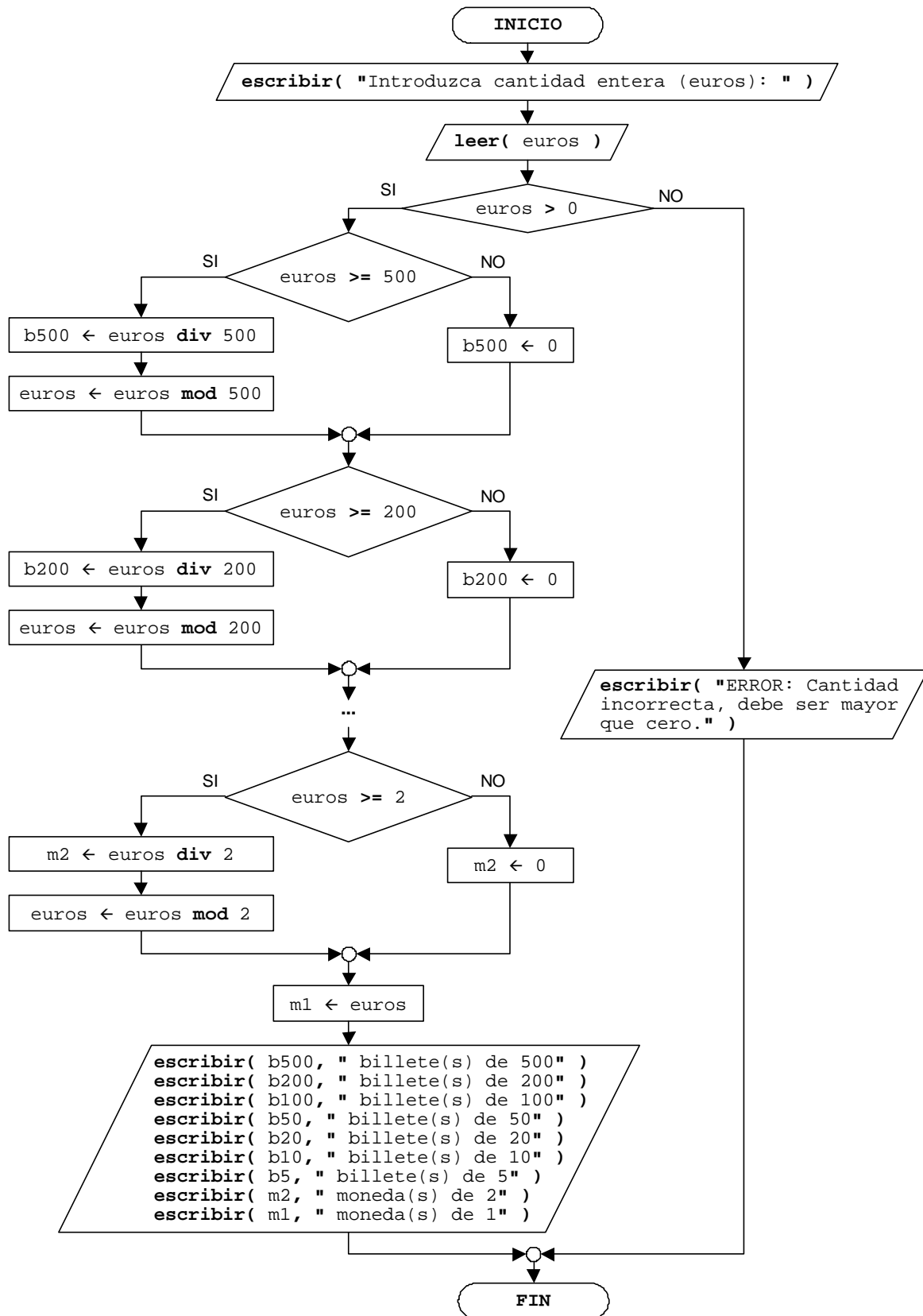
(Pseudocódigo)

Algoritmo Desglose_de_moneda (Solución 1) (ejer_pro_13_23_1.c)	
1	algoritmo Desglose_de_moneda
2	
3	variables
4	entero b500, b200, b100, b50, b20, b10, b5, euros, m2, m1
5	
6	inicio
7	escribir ("Introduzca cantidad entera (euros): ")
8	leer (euros)
9	si (euros > 0)
10	
11	/* Billetes de 500 */
12	si (euros >= 500)
13	b500 ← euros div 500
14	euros ← euros mod 500
15	sino
16	b500 ← 0
17	fin_si
18	
19	/* Billetes de 200 */
20	si (euros >= 200)
21	b200 ← euros div 200
22	euros ← euros mod 200
23	sino
24	b200 ← 0
25	fin_si
26	
27	/* Billetes de 100 */
28	si (euros >= 100)
29	b100 ← euros div 100
30	euros ← euros mod 100
31	sino
32	b100 ← 0
33	fin_si
34	
35	/* Billetes de 50 */
36	si (euros >= 50)
37	b50 ← euros div 50
38	euros ← euros mod 5
39	sino
40	b50 ← 0
41	fin_si
42	
43	/* Billetes de 20 */
44	si (euros >= 20)
45	b20 ← euros div 20
46	euros ← euros mod 20
47	sino
48	b20 ← 0
49	fin_si

```
50
51     /* Billetes de 10 */
52     si ( euros >= 10 )
53         b10 = euros div 10
54         euros = euros mod 10
55     sino
56         b10 = 0
57     fin_si
58
59     /* Billetes de 5 */
60     si ( euros >= 5 )
61         b5 = euros div 5
62         euros = euros mod 5
63     sino
64         b5 = 0
65     fin_si
66
67     /* Monedas de 2 */
68     si ( euros >= 2 )
69         m2 = euros div 2
70         euros = euros mod 2
71     sino
72         m2 = 0
73     fin_si
74
75     /* Monedas de 1 */
76     m1 = euros
77
78     escribir( b500, " billete(s) de 500" )
79     escribir( b200, " billete(s) de 200" )
80     escribir( b100, " billete(s) de 100" )
81     escribir( b50, " billete(s) de 50" )
82     escribir( b20, " billete(s) de 20" )
83     escribir( b10, " billete(s) de 10" )
84     escribir( b5, " billete(s) de 5" )
85     escribir( m2, " moneda(s) de 2" )
86     escribir( m1, " moneda(s) de 1" )
87     sino
88         escribir( "ERROR: Cantidad incorrecta, debe ser mayor que cero." )
89     fin_si
90 fin
```

44 Empezar de cero a programar en lenguaje C

(Ordinograma)



Una segunda **solución** es:

(Pseudocódigo)

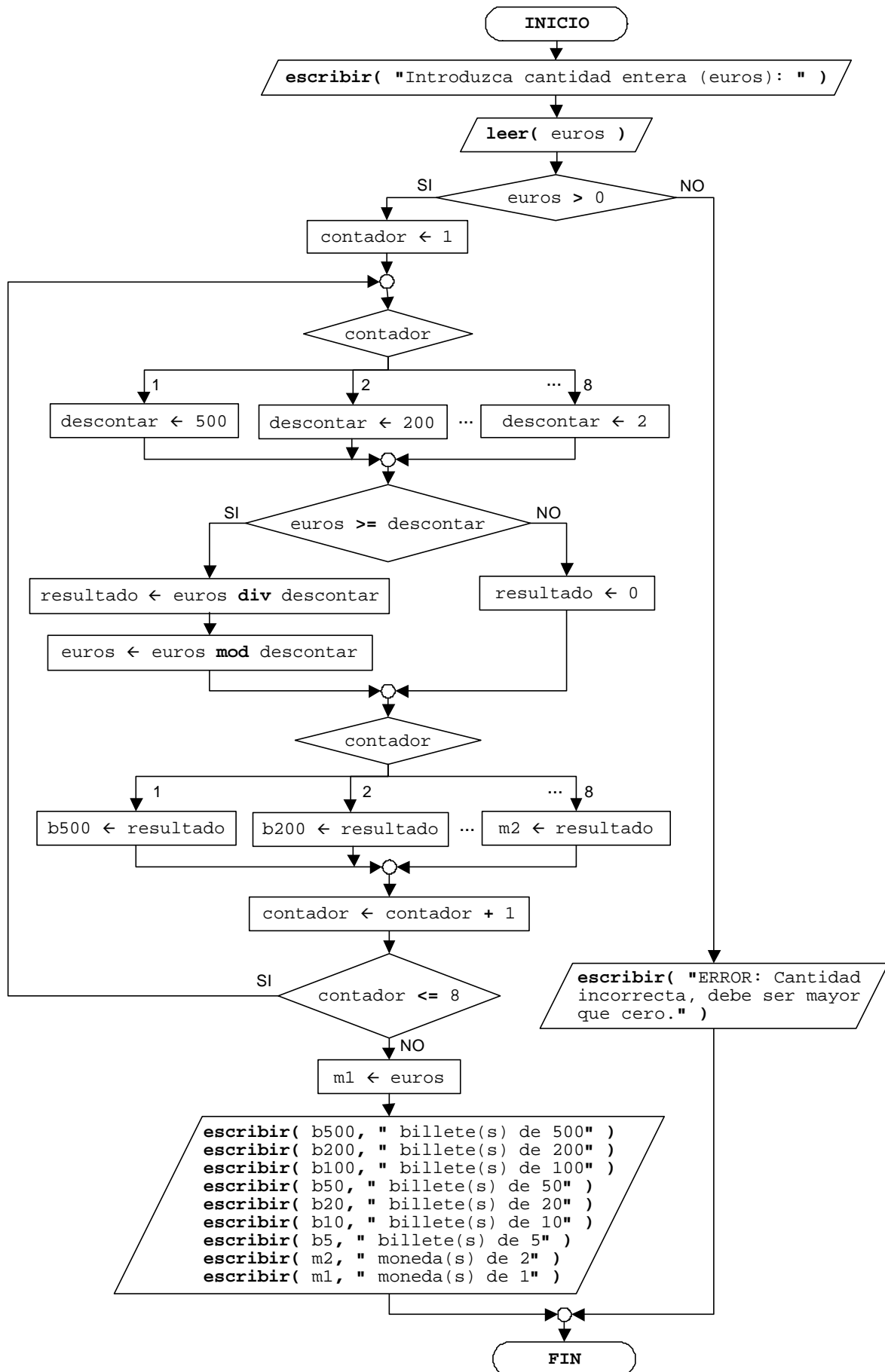
Algoritmo Desglose_de_moneda (Solución 2) (ejer_pro_13_23_2.c)

```

1  algoritmo Desglose_de_moneda
2
3  variables
4      entero b500, b200, b100, b50, b20, b10, b5, contador, descontar,
5          euros, m2, m1, resultado
6
7  inicio
8      escribir( "Introduzca cantidad entera (euros): " )
9      leer( euros )
10     si ( euros > 0 )
11         contador ← 1
12         hacer
13             segun_sea ( contador )
14                 1 : descontar ← 500
15                 2 : descontar ← 200
16                 3 : descontar ← 100
17                 4 : descontar ← 50
18                 5 : descontar ← 20
19                 6 : descontar ← 10
20                 7 : descontar ← 5
21                 8 : descontar ← 2
22             fin_segun_sea
23
24         si ( euros >= descontar )
25             resultado ← euros div descontar
26             euros ← euros mod descontar
27         sino
28             resultado ← 0
29         fin_si
30
31         segun_sea ( contador )
32             1 : b500 ← resultado
33             2 : b200 ← resultado
34             3 : b100 ← resultado
35             4 : b50 ← resultado
36             5 : b20 ← resultado
37             6 : b10 ← resultado
38             7 : b5 ← resultado
39             8 : m2 ← resultado
40         fin_segun_sea
41         contador ← contador + 1
42     mientras ( contador <= 8 )
43
44     m1 ← euros
45
46     escribir( b500, " billete(s) de 500" )
47     escribir( b200, " billete(s) de 200" )
48     escribir( b100, " billete(s) de 100" )
49     escribir( b50, " billete(s) de 50" )
50     escribir( b20, " billete(s) de 20" )
51     escribir( b10, " billete(s) de 10" )
52     escribir( b5, " billete(s) de 5" )
53     escribir( m2, " moneda(s) de 2" )
54     escribir( m1, " moneda(s) de 1" )
55     sino
56         escribir( "ERROR: Cantidad incorrecta, debe ser mayor que cero." )
57     fin_si
58 fin

```

(Ordinograma)



EJERCICIO PROPUESTO 13.24

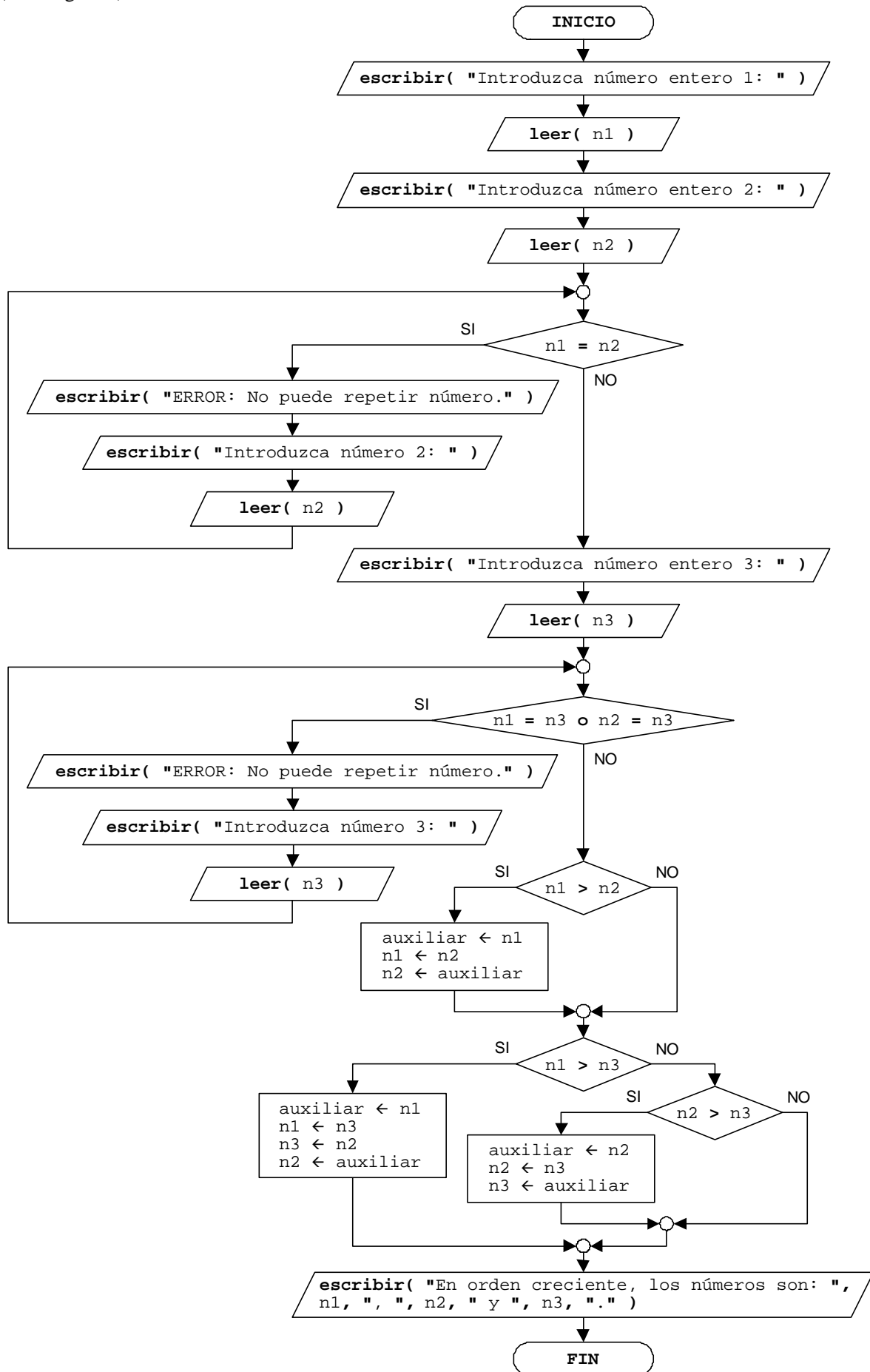
Mostrar tres números en orden creciente

Solución:

(Pseudocódigo)

Algoritmo Mostrar_tres_numeros_en_orden_creciente (ejer_pro_13_24.c)	
1	algoritmo Mostrar_tres_numeros_en_orden_creciente
2	
3	variables
4	entero: auxiliar, n1, n2, n3
5	
6	inicio
7	escribir ("Introduzca número entero 1: ")
8	leer (n1)
9	
10	escribir ("Introduzca número entero 2: ")
11	leer (n2)
12	
13	mientras (n1 = n2)
14	escribir ("ERROR: No puede repetir número.")
15	escribir ("Introduzca número entero 2: ")
16	leer (n2)
17	fin_mientras
18	
19	escribir ("Introduzca número entero 3: ")
20	leer (n3)
21	
22	mientras (n1 = n3 o n2 = n3)
23	escribir ("ERROR: No puede repetir numero.")
24	escribir ("Introduzca número entero 3: ")
25	leer (n3)
26	fin_mientras
27	
28	si (n1 > n2)
29	auxiliar ← n1
30	n1 ← n2
31	n2 ← auxiliar
32	fin_si
33	
34	si (n1 > n3)
35	auxiliar ← n1
36	n1 ← n3
37	n3 ← n2
38	n2 ← auxiliar
39	sino
40	si (n2 > n3)
41	auxiliar ← n2
42	n2 ← n3
43	n3 ← auxiliar
44	fin_si
45	fin_si
46	
47	escribir ("En orden creciente, los números son: ", n1, ", ", n2, " y ",
48	n3, ".")
49	fin

(Ordinograma)



EJERCICIO PROPUESTO 13.25

Solución:

- 1) control repetitiva o iterativa
- 2) verdadera
- 3) iteración
- 4) una
- 5) bucle infinito
- 6) traza
- 7) contador