



**Centro Universitario
Hidalguense**

La sabiduría es nuestra fuerza



FUNDAMENTOS DE PROGRAMACIÓN

**LICENCIATURA EN INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

EXAMEN GLOBAL

PROFESOR:

KARIM RODRIGO ELAYAN NUÑEZ

CHÁVEZ MIRANDA YESSSENIA

Grupo 19

EJERCICIO 1

Un método llamado Ejercicio1 debe retornar verdadero o falso (bool) y recibirá como parámetro un arreglo de enteros (int []) con una longitud entre 1 y 100.

El valor a retornar será verdadero únicamente si el arreglo contiene un numero 7 pero solo si el 7 está junto a otro 7 si hay algún 7 aislado retorna falso.

Por ejemplo:

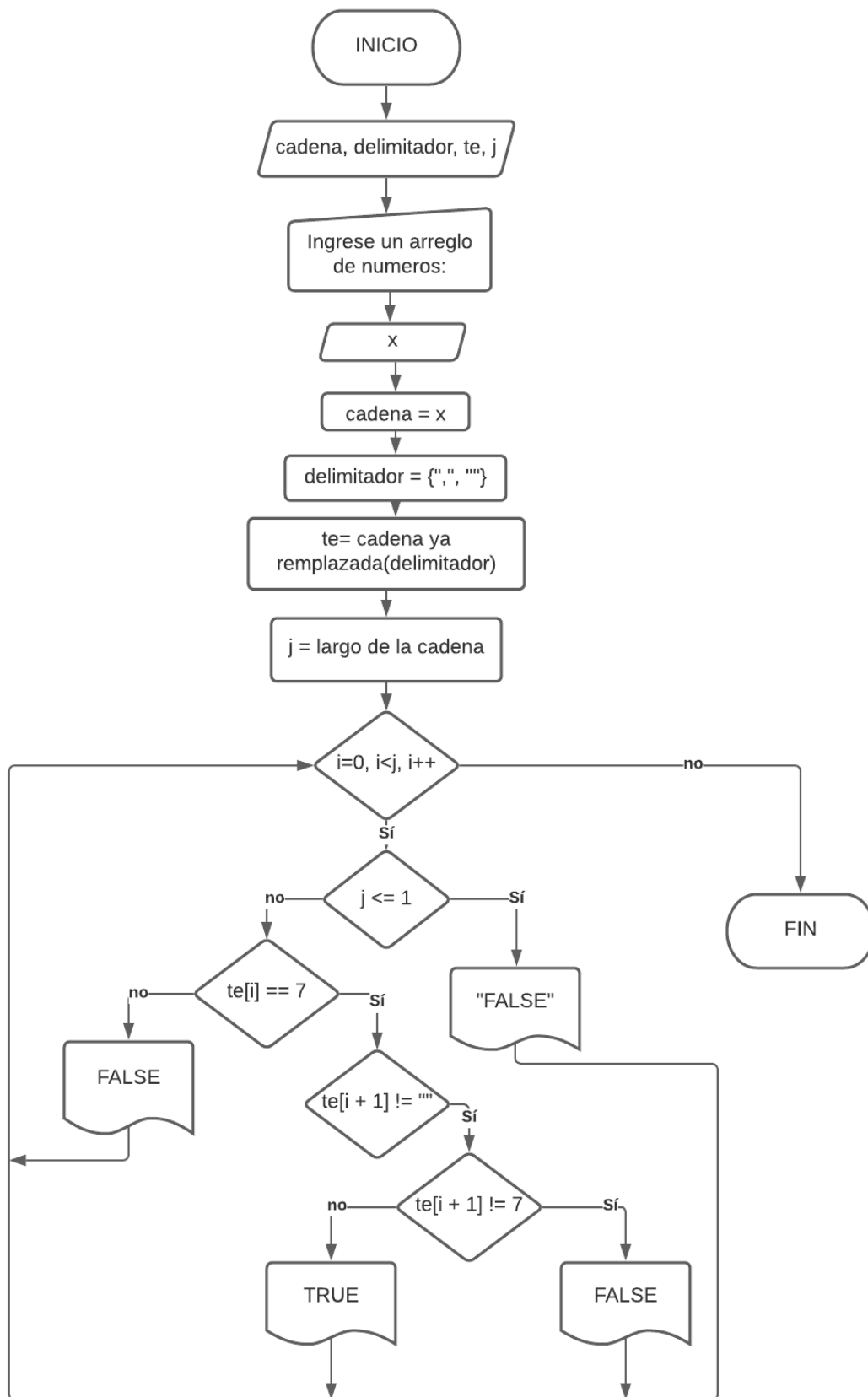
{1, 2, 7, 5, 8} → Falso

{1, 7, 7, 8, 9, 3} → Verdadero

{8, 7, 7, 9, 6, 1, 7} → Falso

Pseudocódigo:

1. Inicio.
2. Ingresar una cadena de números.
3. Leer cadena.
4. Buscar el primer número que sea igual a siete.
5. Comparar si el siguiente número es igual a siete.
6. De lo contrario comparar si anterior es igual es igual a siete.
7. Si en alguno de los dos casos se cumple retornar verdadero.
8. De lo contrario retornar falso.
9. La comparación se repetirá hasta el último dígito de la cadena.
10. Fin.



EJERCICIO 2

Un método llamado Ejercicio2 debe recibir como parámetro un entero entre 1 y 20 y no retorna nada pero al ejecutar el método debe imprimir una pirámide formada con asteriscos '*' y el número que se ingrese determinara la altura de la pirámide.

Por ejemplo

Ejercicio2 (3)

*

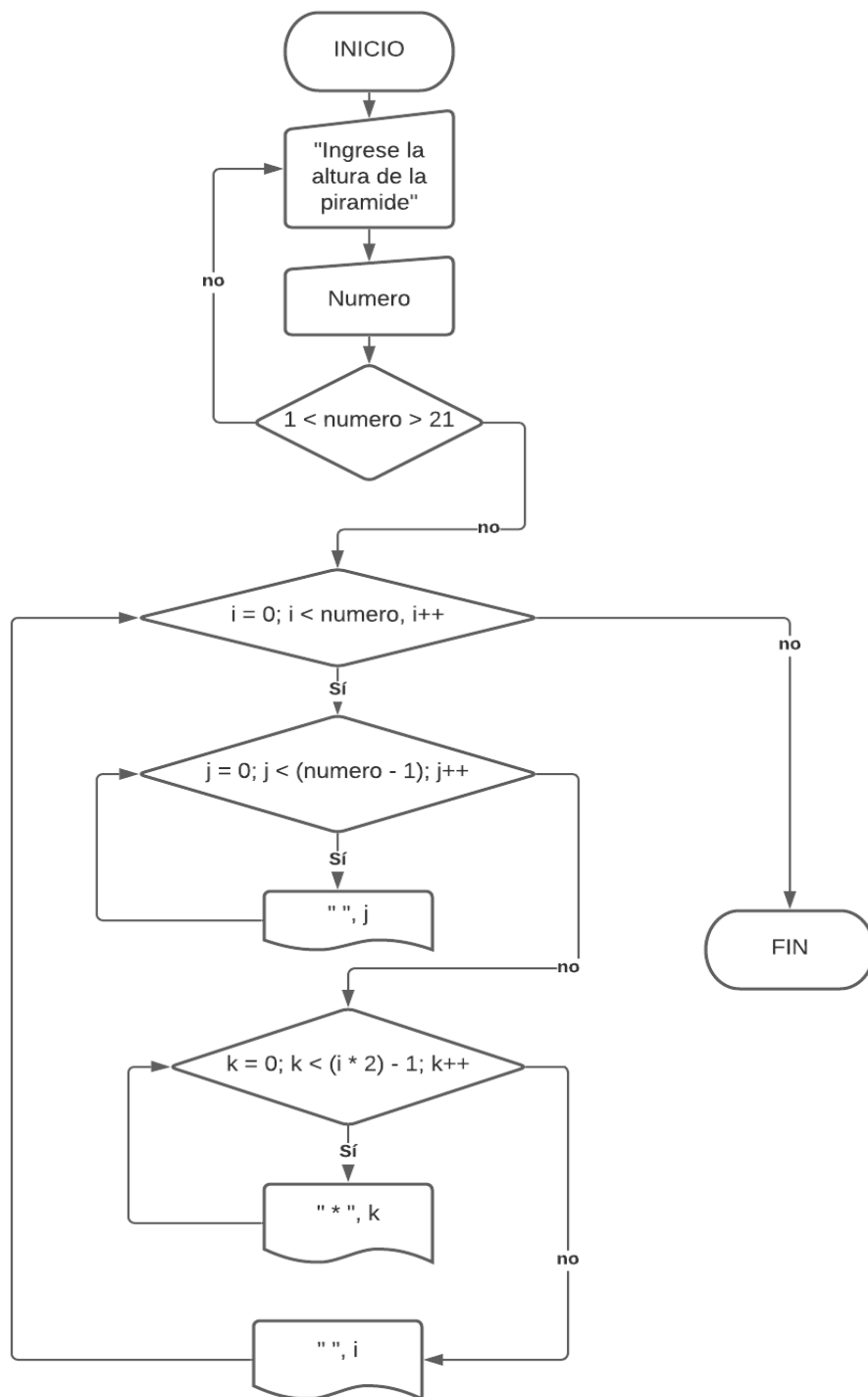
* *

* * *

La pirámide debe tener ese formato de impresión.

Pseudocódigo:

1. Inicio.
2. Ingresar la altura de la pirámide (entre 1 y 20).
3. Guardar valor ingresado en una variable (Numero).
4. Si el numero ingresado es menor a 1 y mayor a 20, entonces:
5. Solicitar que ingrese un valor valido.
6. Si cumple con la condición hacer:
7. Desde $i = 1$ hasta numero hacer
8. Desde $j = 0$ hasta numero -1 hacer
9. Escribir un espacio.
10. Desde $k = 0$ hasta $(i \text{ por } 2) - 1$ hacer
11. Escribir un *
12. Escribir un espacio.
13. Fin.



EJERCICIO 3

Un método llamado Ejercicio3 que recibe un número (float) entre el -99999999.9999 y el 99999999.9999 y debe retornar un flotante.

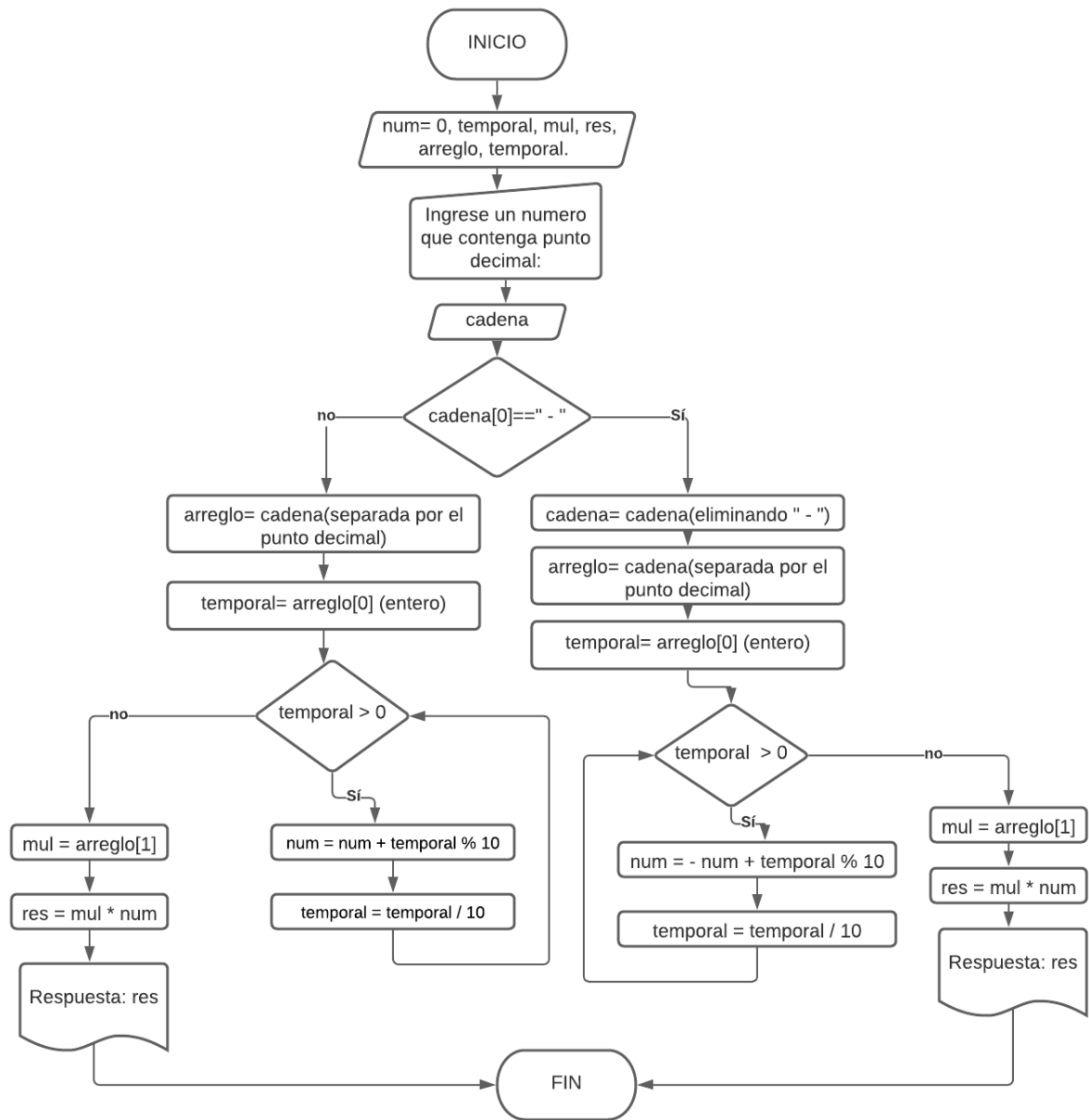
El flotante de retorno se calcula sumando cada dígito individual o en el caso de ser negativo restándolo y multiplicando lo por el número que está después del punto decimal por ejemplo:

{12.5} » $(1+2)*5$ » 15

{-54.2} » $(5-4)*2$ » 2

Pseudocódigo:

1. Inicio.
2. Solicitar que se ingrese un número (con punto decimal) entre el -99999999.9999 y el 99999999.9999.
3. Guardar valor en una cadena.
4. Leer cadena.
5. Separa cadena, de donde se encuentra el punto decimal.
6. Almacenar en nueva cadena (num1), los valores que están antes del punto.(num1)
7. Almacenar en una nueva cadena (num2) los valores que están después del punto.
8. Sumar los valores de la cadena (num1).
9. Guardar en nueva variable (s).
10. Si los valores son negativos, entonces
11. Realizar resta de los valores
12. Guardar en nueva variable (r).
13. Multiplicar valor obtenido por el valor que se encuentra después del punto (num2).
14. Fin



EJERCICIO 4

Un método llamado Ejercicio4 que reciba 4 enteros como las coordenadas de 2 piezas de ajedrez, los rangos son de 1 – 8 y debe retornar un bool indicando si las reinas se pueden atacar o no.

Los primero 2 dígitos corresponden a la primera reina y los siguientes a la segunda reina por ejemplo.

Ejercicio4(3,2,5,4) » verdadero (explicado en la imagen). Recuerden que la reina se puede mover en todas las direcciones y hasta chocar con el borde del tablero.

En las coordenadas (3,2) el 3 representa las filas y el 2 las columnas.

EJERCICIO 5

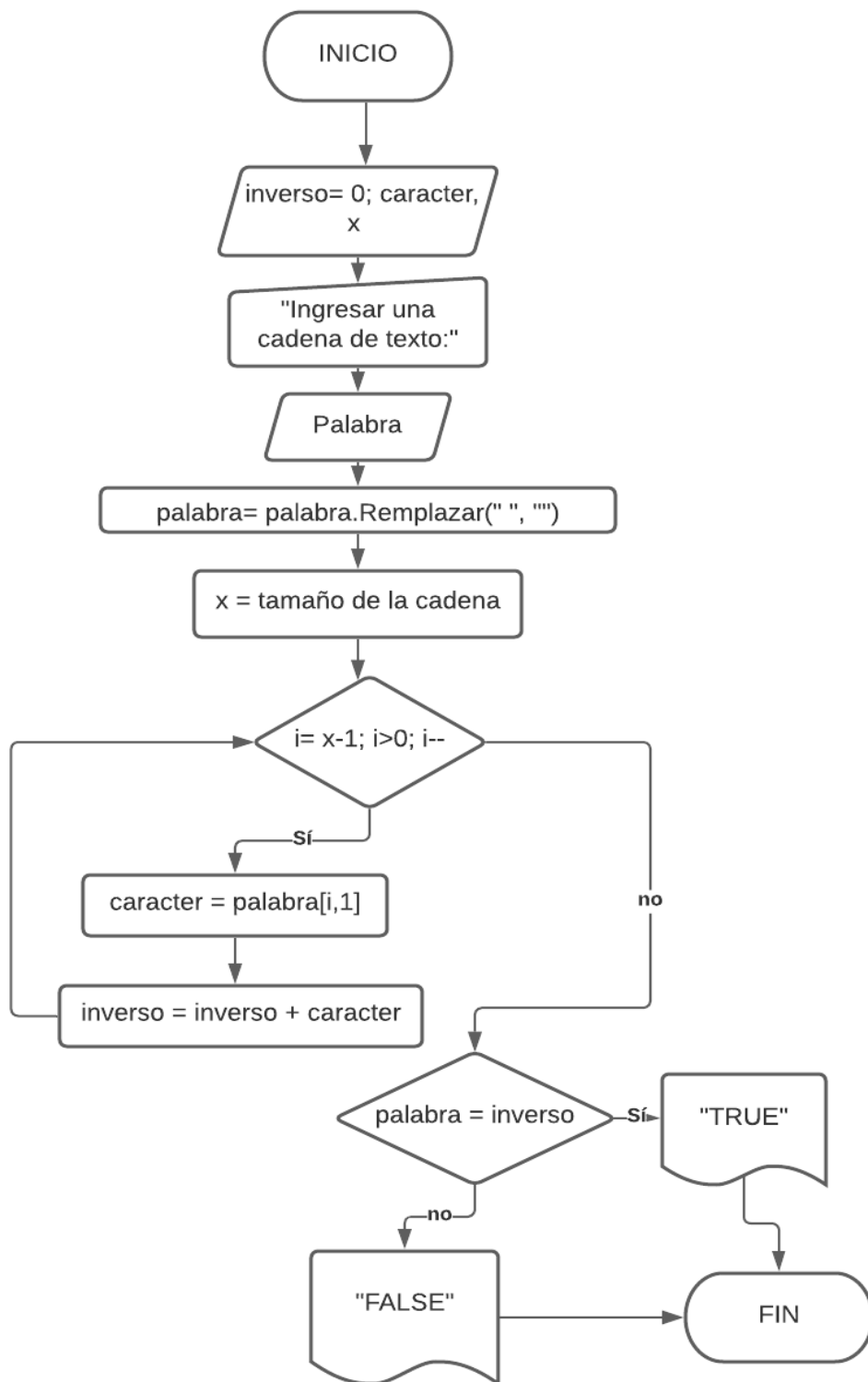
Un método llamado Ejercicio5 que retorna un bool y recibe una cadena de texto debe validar si es un palíndromo o no, debe ignorar mayúsculas o minúsculas así como permitir números.

Por ejemplo, palíndromos validos podrían ser:

- aaaaaaa
- AaAAaAA
- 123321
- ana ana
- a na
- aa

Pseudocódigo:

1. Inicio.
2. Ingresar la cadena de texto.
3. Guardar en variable (palabra).
4. Leer cadena de texto
5. Si hay un espacio en la cadena de texto (" ") entonces
6. Reemplazar (" "), para no tener espacios libres.
7. Leer cadena de texto.
8. Convertir los caracteres de la cadena en mayúsculas.
9. "X" toma el valor del tamaño de la cadena.
10. Declarar variable "inverso".
11. Desde $i = (x - 1)$ hasta 0 hacer;
12. Declarar variable "Carácter".
13. Carácter toma el valor de los caracteres extraídos de una cadena en la posición (i, 1).
14. "Inverso" será igual a inverso + carácter.
15. Si "palabra" es igual a "inverso", entonces;
16. Escribir "TRUE".
17. Si no, entonces;
18. Escribir "FALSE".
19. Fin.



EJERCICIO 6

Un método llamado Ejercicio6 recibe como parámetro un arreglo de enteros con una longitud de entre 1 y 200 y debe retornar un arreglo de enteros de 3 posiciones la primera posición debe indicar el valor más alto, la segunda posición debe indicar el segundo número más alto y la tercera posición debe indicar el número más pequeño del arreglo pasado como parámetro.

Por ejemplo si se ingresa {1,2,3,4,5,6,7,8,9} el arreglo de respuesta debe ser {9,8,1}

Pseudocódigo:

1. Inicio.
2. Crear un arreglo de enteros con una longitud de entre 1 y 200.
3. Declarar variables y, x, z.
4. Ingresar tamaño del arreglo.
5. Guardar en variable "tamaño".
6. Desde i = 0 hasta "tamaño" hacer;
7. Escribir "Ingrese el (1, 2,3...) valor:".
8. Guardar en variable vector en posición i (Vector[i]).
9. Desde i = 0 hasta "tamaño" hacer;
10. Desde j = 0 hasta "tamaño" hacer;
11. Si vector[i] > vector[j] entonces
12. Declarar variable "aux"
13. Aux = vector[i].
14. Vector[i] = vector[j].
15. Vector[j] = aux.
16. Desde i = 0 hasta "tamaño" hacer
17. Declarar variable mayor.
18. Si vector[i] > mayor entonces
19. Mayor = vector[i].
20. Y = mayor.
21. Desde i = 0 hasta "tamaño" hacer
22. Si vector[i] < mayor entonces
23. Mayor = vector[i].
24. X = mayor.
25. Desde i = 0 hasta "tamaño" hacer
26. Si vector[i] > mayor entonces
27. Mayor = vector[i-1]
28. Z = mayor.
29. Escribir "Valor más alto, penúltimo más alto y el menor".
30. Escribir {y , z, x }
31. Fin

