

INSTRUCCIONES DEL EXAMEN

1. Realizar un algoritmo, representándolo mediante pseudocódigo, diagrama de flujo y un programa en C, que:
 - a. Despliegue iterativamente el siguiente menú:
 - i. Mostrar 15 números.
 - ii. Serie descendente
 - iii. Salir.
 - b. Si se elige la opción 1, se debe mostrar en la pantalla 15 números, iniciando de un valor dado por el usuario con un incremento también determinado por el usuario.
 - c. Si se elige la opción 2, se debe solicitar al usuario dos números que indiquen los límites de un rango; se deben imprimir los números (enteros) que se encuentren dentro del rango en orden descendente.

NOTA: La única manera de salir del programa es a través de la opción 3.

ANÁLISIS

- Realizar un algoritmo que muestre 15 opciones en pantalla de forma iterativa:
 - La opción 1 deberá incrementar 15 veces el valor de un número dado por el usuario. Tal incremento también lo deberá proporcionar el usuario.
 - En la opción 2 se le solicitará al usuario que ingrese dos números para determinar el rango. Posteriormente se le solicitará un tercer número para evaluar si se encuentra dentro del rango que proporcionó. Si el número se encuentra dentro de ese rango, se deberá imprimir en decremento los números que se encuentren dentro del rango, a partir del número que ingresa el usuario. Y si no se encuentra dentro de ese rango, se deberá imprimir un mensaje, comunicándole al usuario que tal número no está dentro del rango determinado.
 - La opción 3 saldrá del programa.

OBJETIVO

- Mostrar en pantalla las 15 opciones de forma iterativa.
- Dado el valor que ingresará el usuario, se deberá incrementar 15 veces el valor de un número que proporcione el usuario.

- Evaluar si el número que ingresa el usuario, se encuentra dentro del rango que proporcionará el mismo usuario. Y en caso de que se encuentre, deberá ir en decremento tal número y mostrar en pantalla los números que están dentro del rango.
- Salir del programa con la opción 3.

ENTRADAS

- **Menú:**
 - eNumber: variable de tipo entero que almacenará la opción que escoja el usuario.
- **Opción 1:**
 - eNumberIncrement: variable de tipo entero que almacenará el digito que ingrese el usuario para incrementar 15 veces su valor.
 - eNumberBase: variable de tipo entero que almacenará el digito base que ingrese el usuario para incrementar su valor con el valor de la variable anterior.
- **Opción 2:**
 - eFirstLimit: variable de tipo entero que almacenará uno de los límites del rango.
 - eSecondLimit: variable de tipo entero que almacenará otro de los límites del rango.
 - eNumberRange: variable de tipo entero que almacenará el digito a evaluar.

PRE-CONDICIONES

- **Menú:**
 - sOptions[15][15] será un arreglo de cadenas que almacenará las 15 opciones del menú.
 - eCounterOptions será una variable contador que incrementará en 1 su valor para mostrar en pantalla las 15 opciones de forma iterativa.
 - eLength será una variable de tipo entero que almacenará la longitud del arreglo.
- **Opción 1:**
 - eCounterIncrement será una variable contador que incrementará 15 veces su valor para que muestre 15 números.
- **Opción 2:**
 - eHigherLimit será una variable de tipo entero que almacenará el valor del límite más grande.

- eLessLimit será una variable de tipo entero que almacenará el valor del límite más pequeño.
- Para resolver estos problemas, deberá usarse los ciclos: for, do while, while y las estructuras de control: if, if else y switch.
- Salir del programa hasta que ingrese la opción 3.

RESTRICCIONES

- No se deberá ingresar valores de tipo char o alfanuméricos.
- Si no se ingresa la opción 3, el programa debe seguir ejecutándose.

SALIDA

- Se deberán mostrar las 15 opciones del menú.
- Si escoge la opción 1, se imprimirá en pantalla: eNumberBase
- Si escoge la opción 2, se imprimirá en pantalla: eNumber

PSEUDOCÓDIGO

o. Inicio

1. Inicializar variables:

1.1 Menú

1.1.1 sOptions[15][15] \leftarrow {"Opción 1", "Opción 2", "Opción 3", "Opción 4", "Opción 5", "Opción 5", "Opción 7", "Opción 8", "Opción 9", "Opción 10", "Opción 11", "Opción 12", "Opción 13", "Opción 14", "Opción 15"}

1.1.2 eNumber \leftarrow 0

1.1.3 eCounterOptions \leftarrow 0

1.1.4 eLength \leftarrow 0

1.2 Opción 1

1.2.1 eNumberBase \leftarrow 0

1.2.2 eNumberIncrement \leftarrow 0

1.2.3 eCounterIncrement \leftarrow 1

1.3 Opción 2

- 1.3.1 $eFirstLimit \leftarrow 0$
 - 1.3.2 $eSecondLimit \leftarrow 0$
 - 1.3.3 $eNumberRange \leftarrow 0$
 - 1.3.4 $eHigherLimit \leftarrow 0$
 - 1.3.5 $eLessLimit \leftarrow 0$
- 2. $eLength \leftarrow \text{sizeof}(sOptions) / \text{sizeof}(sOptions[0])$
- 3. Para($eCounterOptions$; $eCounterOptions \leq eLength$; $eCounterOptions++$)
 - 3.1 Imprimir $sOptions[eCounterOptions]$
- 4. Fin Para.
- 5. Hacer
 - 5.1 Imprimir “Escoge una opción: ”
 - 5.2 Leer $eNumber$.
 - 5.3 Caso($eNumber$)
 - 5.3.1 caso 1:
 - 5.3.1.1 Imprimir “Escribe un número: ”
 - 5.3.1.2 Leer $eNumberBase$.
 - 5.3.1.3 Imprimir “Escribe otro número para incrementar el anterior: ”
 - 5.3.1.4 Leer $eNumberIncrement$.
 - 5.3.1.5 Mientras($eCounterIncrement \leq 15$)
 - 5.3.1.5.1 Imprimir $eNumberBase$
 - 5.3.1.5.2 $eNumberBase += eNumberIncrement$
 - 5.3.1.5.3 $eCounterIncrement++$
 - 5.3.1.6 Fin Mientras
 - 5.3.2 Fin caso 1.
 - 5.3.3 caso 2:
 - 5.3.3.1 Imprimir “Escribe un número límite para determinar el rango: ”
 - 5.3.3.2 Leer $eFirstLimit$.

5.3.3.3 Imprimir “Escribe otro número límite para determinar el rango: ”

5.3.3.4 Leer eSecondLimit.

5.3.3.5 Imprimir “Escribe un número para evaluar que se encuentre dentro del rango establecido por los números que ingresaste anteriormente: ”

5.3.3.6 Leer eNumberRange.

5.3.3.7 Si(eFirstLimit < eSecondLimit) entonces

5.3.3.7.1 eLeesLimit \leftarrow eFirstLimit

5.3.3.7.2 eHigherLimit \leftarrow eSecondLimit

5.3.3.8 Fin Si.

5.3.3.9 Si no

5.3.3.9.1 eLeesLimit \leftarrow eSecondLimit

5.3.3.9.2 eHigherLimit \leftarrow eFirstLimit

5.3.3.10 Fin Si no

5.3.3.11 Hacer

5.3.3.11.1 Imprimir eNumberRange.

5.3.3.11.2 eNumberRange - -

5.3.3.12 Fin Hacer

5.3.3.13 Mientras(eNumberRange > eLessLimit && eNumberRange < eHigherLimit)

5.3.3.14 Fin Mientras

5.3.4 Fin caso 2

5.3.5 caso 3:

5.3.5.1 Imprimir “El programa ha terminado.”

5.3.6 Fin caso 3.

5.3.7 default

5.3.7.1 Imprimir “Sólo se permiten las opciones 1, 2 y 3.\n”

5.3.8 Fin default.

5.4 Fin Caso.

6. Fin Hacer.

7. Mientras(eNumber != 3)
8. Fin Mientras
9. Fin.

DIAGRAMA DE FLUJO



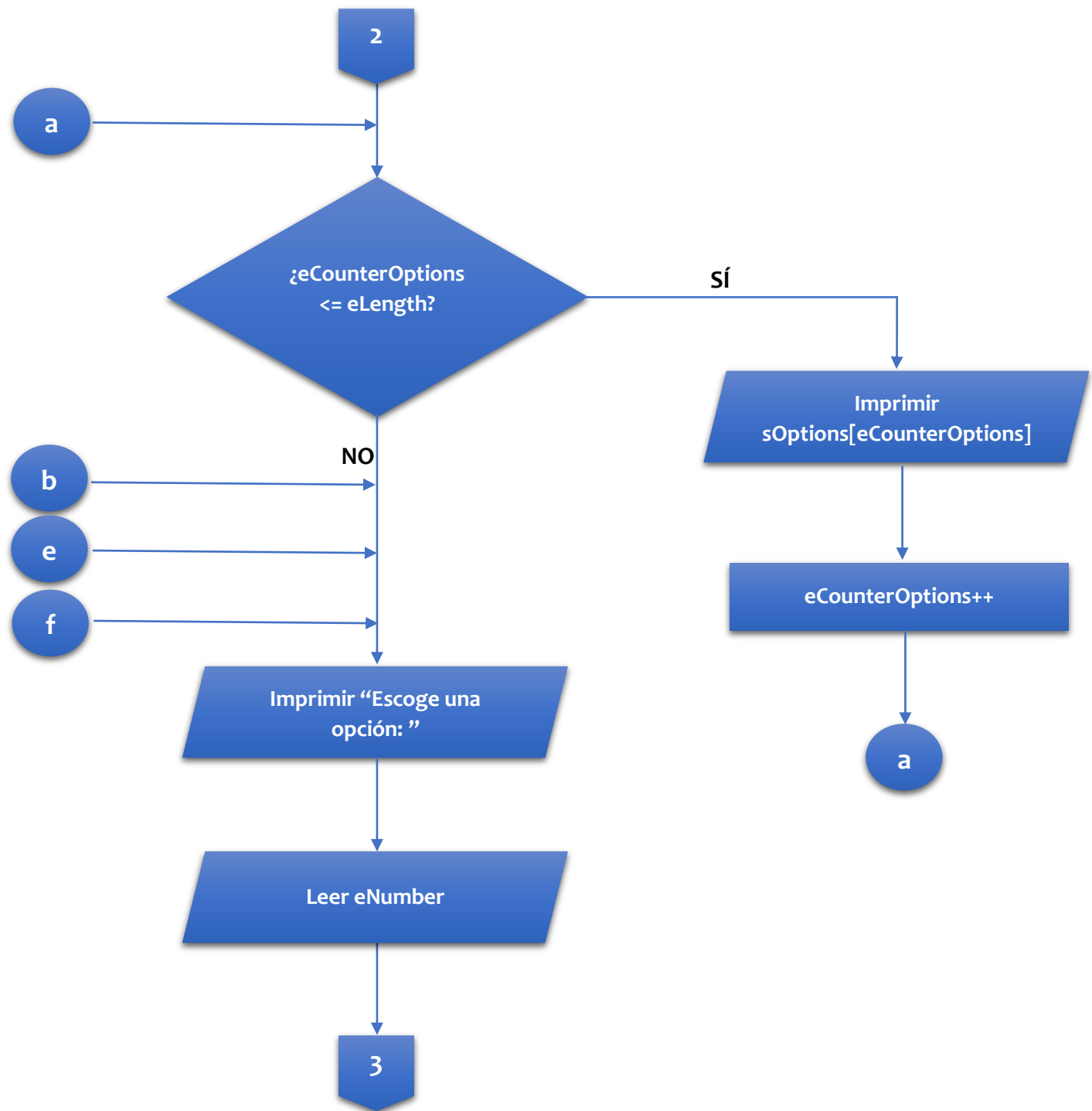
1

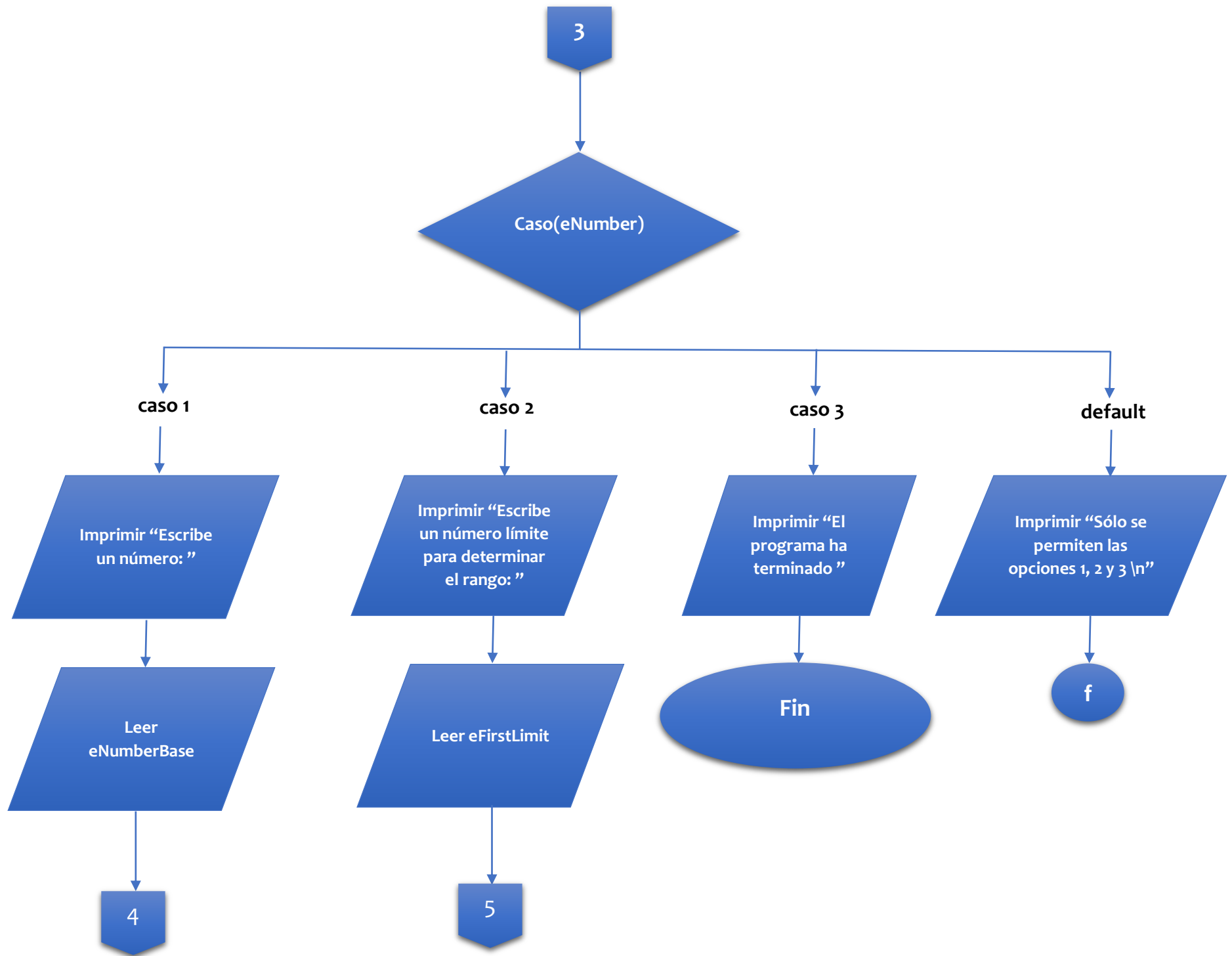
Inicializar variables:

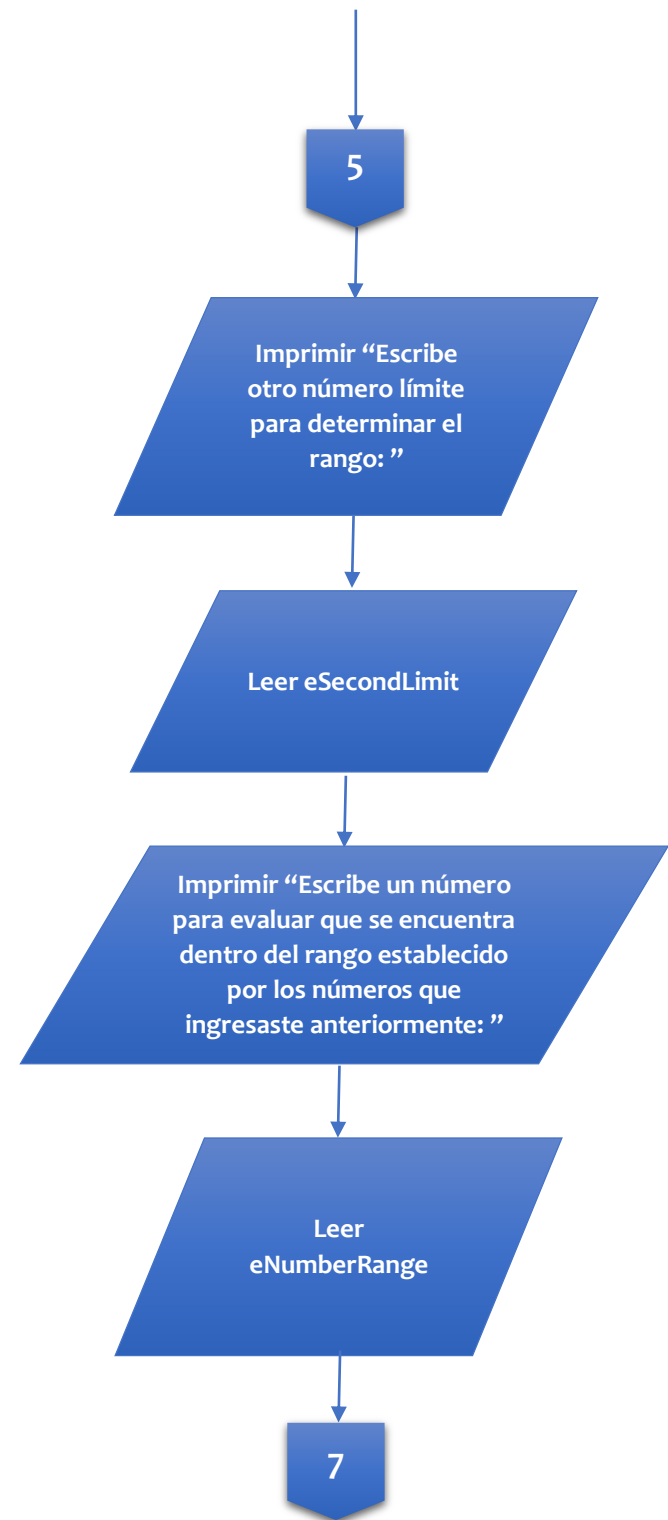
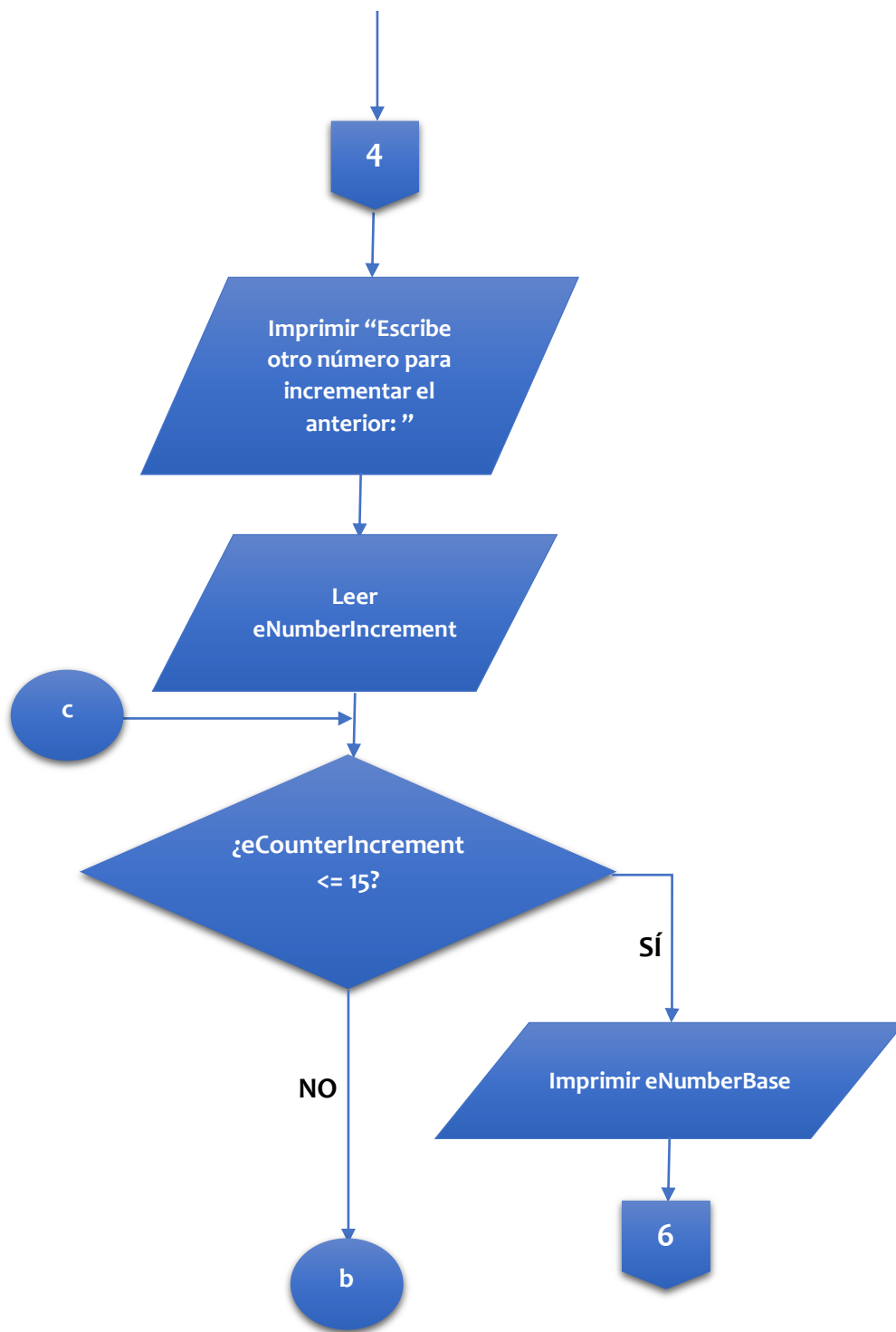
- $sOptions[15][15] \leftarrow \{ \text{"Opción 1"}, \text{"Opción 2"}, \text{"Opción 3"}, \text{"Opción 4"}, \text{"Opción 5"}, \text{"Opción 5"}, \text{"Opción 7"}, \text{"Opción 8"}, \text{"Opción 9"}, \text{"Opción 10"}, \text{"Opción 11"}, \text{"Opción 12"}, \text{"Opción 13"}, \text{"Opción 14"}, \text{"Opción 15"} \}$
 - $eNumber \leftarrow 0$
 - $eCounterOptions \leftarrow 0$
 - $eLength \leftarrow 0$
 - $eNumberBase \leftarrow 0$
 - $eNumberIncrement \leftarrow 0$
 - $eCounterIncrement \leftarrow 1$
 - $eFirstLimit \leftarrow 0$
 - $eSecondLimit \leftarrow 0$
 - $eNumberRange \leftarrow 0$
 - $eHigherLimit \leftarrow 0$
 - $eLessLimit \leftarrow 0$

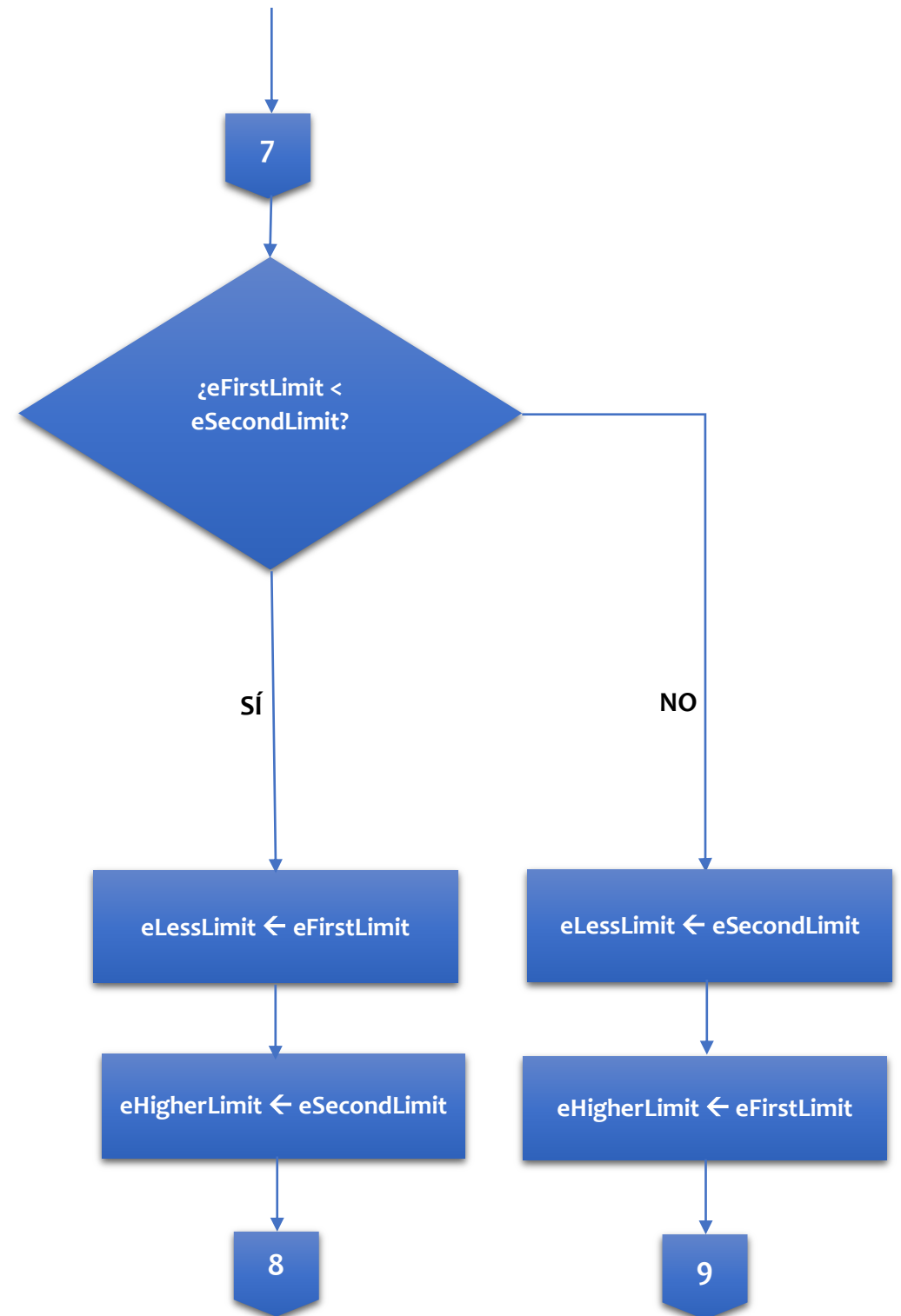
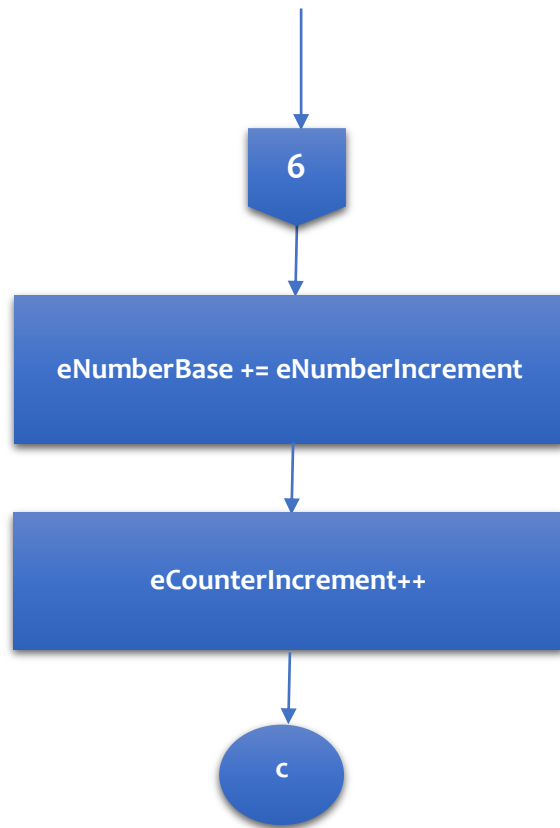
$eLength = \text{sizeof}(sOptions) / \text{sizeof}(sOptions[0])$

2









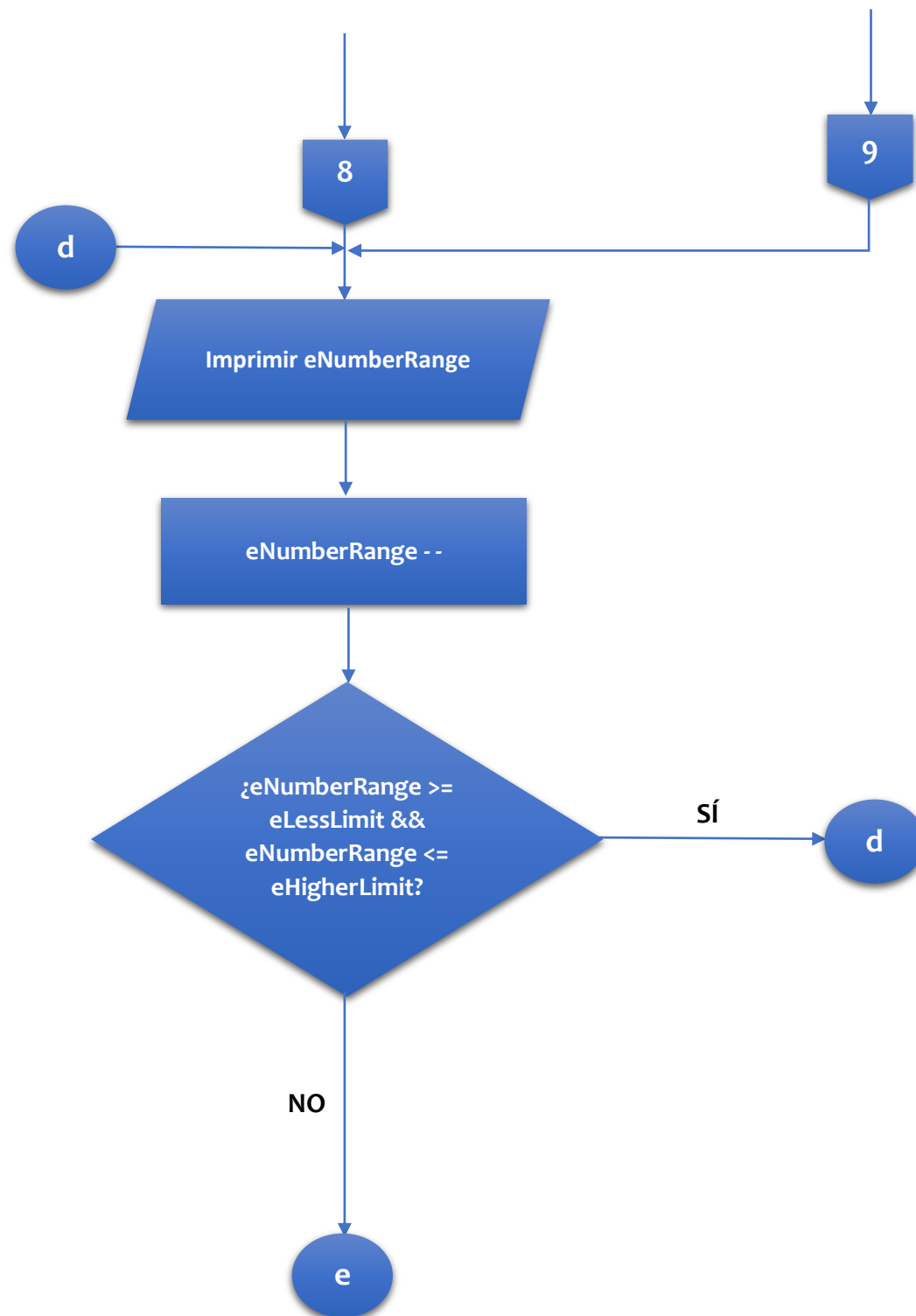


DIAGRAMA N-S

Inicio

Entradas:

- eNumber: entero
- eNumberIncrement: entero
- eNumberBase: entero
- eFirstLimit: entero
- eSecondLimit: entero
- eNumberRange: entero

Inicializar variables:

- sOptions[15][15] \leftarrow {"Opción 1", "Opción 2", "Opción 3", "Opción 4", "Opción 5", "Opción 5", "Opción 7", "Opción 8", "Opción 9", "Opción 10", "Opción 11", "Opción 12", "Opción 13", "Opción 14", "Opción 15"}
- eNumber \leftarrow 0
- eCounterOptions \leftarrow 0
 - eLength \leftarrow 0
- eNumberBase \leftarrow 0
- eNumberIncrement \leftarrow 0
- eCounterIncrement \leftarrow 1
 - eFirstLimit \leftarrow 0
 - eSecondLimit \leftarrow 0
 - eNumberRange \leftarrow 0
 - eHigherLimit \leftarrow 0
 - eLessLimit \leftarrow 0

eLength \leftarrow sizeof(sOptions) / sizeof(sOptions[0])

Para eCounterOptions hasta eCounterOptions \leq eLength incrementar eCounterOptions++

Imprimir sOptions[eCounterOptions]

Fin Para

Mientras(eNumber != 3)	Hacer						
	Imprimir “Escoge una opción: ”						
	Leer eNumber.						
	Caso(eNumber)						
	caso 1	caso 2	caso 3	default			
	Imprimir “Escribe un número: ”	Imprimir “Escribe un número límite para determinar el rango: ”	Imprimir “El programa ha terminado.”	Imprimir “Sólo se permiten las opciones 1, 2 y 3.”			
	Leer eNumberBase	Leer eFirstLimit					
	Imprimir “Escribe otro número para incrementar el anterior: ”	Imprimir “Escribe otro número límite para determinar el rango: ”					
	Leer eNumberIncrement	Leer eSecondLimit					
	Mientras(eCounterIncrement <= 15)	Imprimir “Escribe un número para evaluar que se encuentre dentro del rango establecido por los números que ingresaste anteriormente: ”					
		Leer eNumberRange					
		¿eFirstLimit < eSecondLimit?					
	eCounterIncrement++		eLessLimit ← eFirstLimit		eLessLimit ← eSecondLimit		

			<div>eHigherLimit ← eSecondLimit</div> <div>eHigherLimit ← eFirstLimit</div> <div>Hacer</div> <div> <div>Imprimir eNumberRange</div> <div>eNumberRange - -</div> </div> <div>Mientras(eNumberRange >= eLessLimit && eNumberRange <= eHigherLimit)</div>		
Fin					

PRUEBA DE ESCRITORIO							
Prueba	Datos de entrada						Salida
	eNumber	eNumberBase	eNumberIncrement	eFirstLimit	eSecondLimit	eNumberRange	Variables
a	1	5	1				5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
b	2			10	20	15	15 14 13 12 11 10
c	3						El programa ha terminado.
d	10						Sólo se permiten opciones 1, 2 y 3.
e	3						El programa ha terminado.