**ÁNALISIS**

* Pedir al usuario que ingrese el límite inferior, el límite superior de un rango y verificar que el tercer número ingresado esté dentro del rango.
* Pedir los 3 números al usuario hasta que ingrese un número que no se encuentra dentro del rango.

**OBJETIVO**

* Validar que un número dado por el usuario se encuentre en un rango determinado y no salir del bucle hasta que el usuario ingrese un número que no encuentre dentro del rango.

**ENTRADAS**

* **eNumber:** variable de tipo entero que almacenará el valor del número que se encontrará dentro o fuera del rango establecido.
* **eFirstNumber:** variable de tipo entero que determinará el límite inferior o superior del rango.
* **eSecondNumber:** variable de tipo entero que determinará el otro límite inferior o superior del rango.

**PRE-CONDICIONES**

* eLowerLimit es una variable de tipo entero que almacenará el límite inferior. Esta variable guardará el valor de las variables de entrada (eFirstNumber o eSecondNumber), dependiendo cuál de las dos tenga el valor más pequeño.
* sHigherLimir es una variable de tipo entero que almacenará el límite superior. Esta variable guardará el valor de las variables de entrada (eFirstNumber o eSecondNumber), dependiendo cuál de las dos tienga el valor más grande.
* Se deberá usar el bucle do-while para realizar este algoritmo.
* Se debe mantener en el bucle hasta que el usuario ingrese un número que NO se encuentre dentro del rango establecido.

**RESTRICCIONES**

* No se aceptarán valores alfanuméricos ni decimales.

**SALIDA**

* Si el usuario ingresa un número dentro del rango, imprimir: eNumber “sí está dentro del rango establecido.”
* Si el usuario ingresa un número que no se encuentra dentro del rango, imprimir: eNumber “no está dentro del rango establecido.”

**PSEUDOCÓDIGO**

1. Inicio
2. Inicializar variables:
   1. eNumber 🡨 0
   2. eFirstNumber 🡨 0
   3. eSecondNumber 🡨 0
   4. eLowerLimit 🡨 0
   5. eHigherLimit 🡨 0
3. Imprimir “Ingresa un número límite para especificar el rango: ”
4. Leer eFirstNumber.
5. Imprimir “Ingresa otro número límite para especificar el rango: ”
6. Leer eSecondNumber.
7. Hacer
   1. Imprimir “Ingresa cualquier número para validar que se encuentre dentro del rango establecido: ”
   2. Leer eNumber.
   3. Si(eFirstNumber < eSecondNumber) entonces
      1. eLowerLimit 🡨 eFirstNumber
      2. eHigherLimit 🡨 eSecondNumber
   4. Fin Si
   5. Si no
      1. eLowerLimit 🡨 eSecondNumber
      2. eHigherLimit 🡨 eFirstNumber
   6. Fin Si no
   7. Si(eNumber < eLowerLimit || eNumber > eHigherLimit) entonces
      1. Imprimir**:** eNumber “no se encuentra dentro del rango establecido.”
   8. Fin Si.
   9. Si no
      1. Imprimir: eNumber “se encuentra dentro del rango establecido.”
   10. Fin Si no
8. Fin Hacer
9. Mientras(eNumber >= eLowerLimit && eNumber <= eHigherLimit)
10. Fin Mientras
11. Fin

**DIAGRAMA DE FLUJO**

**Entradas:**

* **eNumber: entero**
* **eFirstNumber: entero**
* **eSecondNumber: entero**

**Inicializar variables:**

* **eNumber 🡨 0**
* **eFirstNumber 🡨 0**
* **eSecondNumber 🡨 0**
* **eLowerNumber 🡨 0**
* **eHigherNumber 🡨 0**

**Imprimir “Ingresa un número límite para especificar el rango: ”**

**Leer eFirstNumber**

**1**

**1**

**Imprimir “Ingresa otro número límite para especificar el rango**: **”**

**Leer eSecondNumber**

**Imprimir “Ingresa cualquier número para validar que se encuentre dentro del rango establecido: ”**

**Leer eNumber**

**¿eFirstNumber < eSecondNumber?**

**No**

**Sí**

**3**

**2**

**3**

**2**

**eLowerLimit = eSecondNumber;**

**eLowerLimit = eFirstNumber;**

**eHigherLimit = eFirstNumber;**

**eHigherLimit = eSecondNumber;**

**¿eNumber < eLowerLimit || eNumber > eHigherLimit?**

**No**

**Sí**

**Imprimir: eNumber “se encuentra dentro del rango establecido.”**

**Imprimir: eNumber “no se encuentra dentro del rango establecido.”**

|  |  |  |  |
| --- | --- | --- | --- |
| DIAGRAMA N-S | | | |
| Inicio | | | |
| Entradas:   * eNumber: entero * eFirstNumber: entero * eSecondNumber: entero | | | |
| Inicizalizar variables:   * eNumber 🡨 0 * eFirstNumber 🡨 0 * eSecondNumber 🡨 0 * eLowerNumber 🡨 0 * eHigherNumber 🡨 0 | | | |
| Imprimir “Ingresa un número límite para especificar el rango: ” | | | |
| Leer eFirstNumber | | | |
| Imprimir “Ingresa otro número límite para especificar el rango: ” | | | |
| Leer eSecondNumber | | | |
|  | **Imprimir “Ingresa cualquier número para validar que se encuentre dentro del rango establecido: ”** | | |
| **Leer eNumber**  **¿eFirstNumber < eSecondNumber?** | | |
| **No**  **Sí** | | |
| **eLowerLimit 🡨 eFirstNumber;** | **eLowerLimit 🡨 eSecondNumber** | |
| **eHigherLimit 🡨 eSecondNumber** | **eHigherLimit 🡨 eFirstNumber** | |
| **No**  **Sí**  **¿eNumber < eLowerLimit || eNumber > eHigherLimit?** | | |
| **Imprimir: eNumber “no se encuentra dentro del rango establecido.”** | | **Imprimir: eNumber “se encuentra dentro del rango establecido.”** |
| Mientras(eNumber >= eLowerLimit && eNumber <= eHigherLimit) | | | |
| Fin | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PRUEBA DE ESCRITORIO | | | | |
| Prueba | **Datos de entrada** | | | **Salida** |
| **eFirstNumber** | **eSecondNumber** | **eNumber** | **eNumber** |
| a | **1** | **10** | **11** | **11 no se encuentra dentro del rango establecido.** |
| b | **10** | **5** | **8** | **8 se encuentra dentro del rango establecido.** |
| c | **20** | **40** | **2** | **2 no se encuentra dentro del rango establecido.** |
| d | **80** | **10** | **75** | **75 se encuentra dentro del rango establecido.** |