

Ejercicio resuelto de la Práctica 2: Ciclo Mientras, Bandera o flag para búsqueda de mínimo

19-CONSTRUIR UN PROGRAMA QUE PUEDA INGRESAR VARIOS NUMEROS ENTEROS, DISTINTOS DE CERO, DE UNO POR VEZ. FINALIZA EL INGRESO DE LOS DATOS AL LEER UN VALOR NULO. INFORMAR: A) LA CANTIDAD DE VALORES INGRESADOS COMPRENDIDOS ENTRE π y 3π . B) EL MENOR VALOR INGRESADO.

Pasos para la resolución: Como buena práctica, sugerimos hacer el diseño de la estrategia:

1) Análisis del problema (QUE??): Leer detenidamente el enunciado, e identificar:

Datos: Cuáles son los datos del problema: Conozco varios números enteros. Ojo! Los números enteros pueden ser positivos o negativos distintos de cero! Porque cero lo utilizamos para el fin del ciclo.

Proceso: Qué operaciones debo hacer: Tengo que contar cuántos números son mayores o iguales a 3.14 (π) y menores o iguales a $3*3.14$ (3π). Además, tengo que hallar el mínimo valor entre todos los números.

¿Cuántos números son? No lo sé. Pero puedo identificar que terminarán con $\text{numero}=0$.

Resultados: Qué resultados me piden: Mostrar la cantidad de números dentro del rango mencionado y el valor mínimo entre todos los números.

2) Diseño de la estrategia (COMO??): Defino cómo voy a diseñar el algoritmo, y le doy un orden lógico a las tareas que voy a realizar:

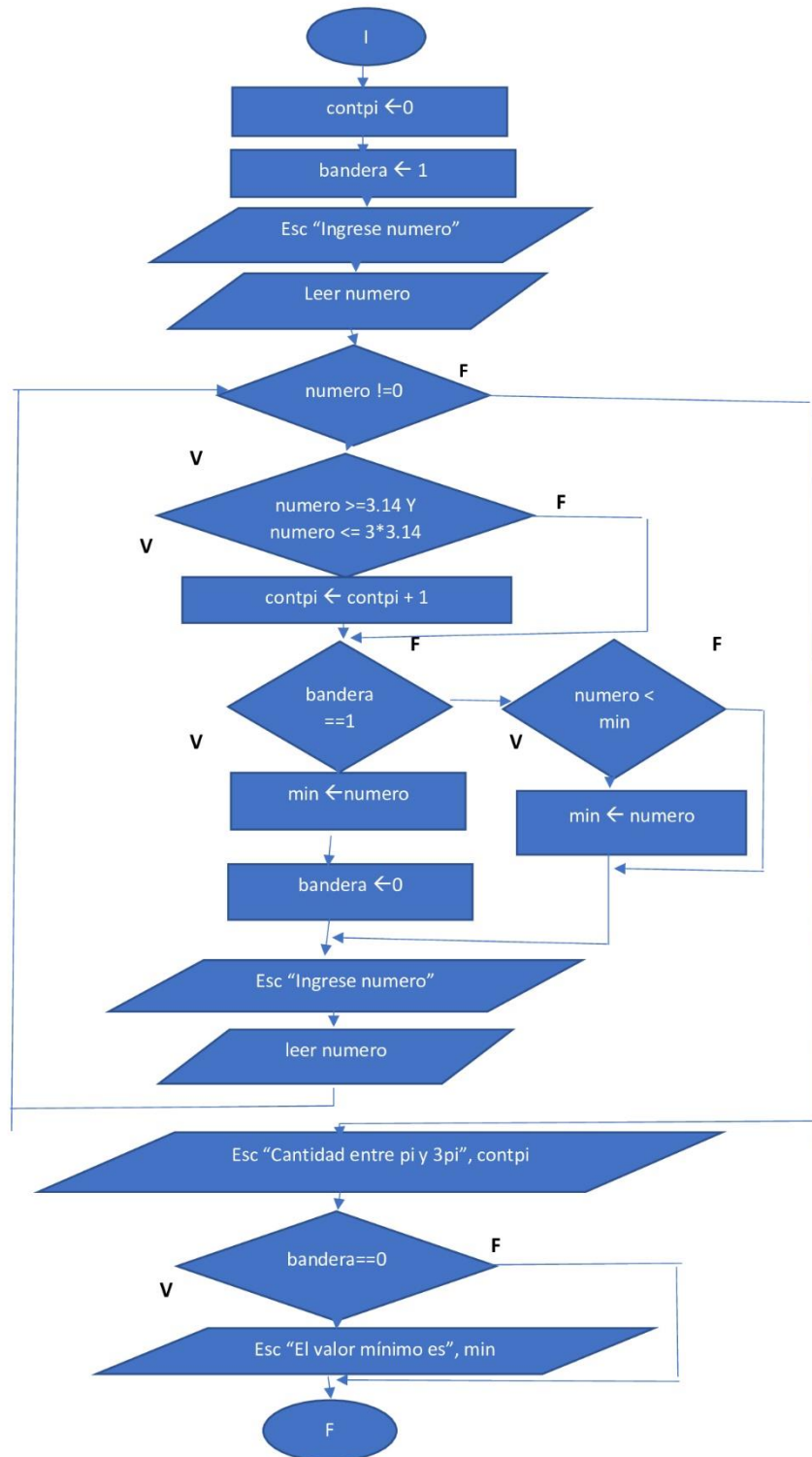
a) **Entradas:** Ingreso cada número entero, haciendo **entrada de datos** en la variable: **numero**.

b) **Proceso:** Primero identifico la **condición de fin**: Mientras el número sea distinto de cero voy a procesar un número. **Procesar cada número significa**: voy a sumar 1 a un contador de números si está dentro del rango mencionado y voy a ir calculando el valor mínimo parcial. **El proceso lo hago para un número, luego lo repito con una sentencia de repetición.**

c) **Salida:** Informo: **Dónde: Al finalizar el proceso de cada número**, informo la cantidad de números comprendidos entre π y 3π y el valor mínimo de esos números.

Identificamos también, qué variables vamos a utilizar, y las enumeramos con su tipo:
En este ejercicio las variables serán: **numero**, **contpi**, **bandera**, **min** tipo entero.

3) Diseño el algoritmo en base a la estrategia en Diagrama de Flujo:



a) La variable `contpi` se inicializaron en cero fuera del ciclo de repetición, que es dónde la voy a usar. ¿Para qué? Las variables en la memoria no tienen valor inicial. Yo le asigno un valor inicial. La primera vez que ejecuto la sentencia `contpi ← contpi + 1`, a la variable `contpi` le asigno lo que tiene en ese momento la variable `contpi` (que la primera vez es 0) más 1. La vez siguiente ya tiene un valor definido, que es 1, y al sumarle 1 quedará el valor 2 y así sucesivamente. **Siempre que necesite contar o acumular valores, la variable que utilizo de contador o acumulador debe ser inicializada en 0 antes del proceso donde la utilizo.**

a) bandera ← 1: **Voy a utilizar una variable tipo bandera o flag para poder hacer, por única vez, la asignación del primer número a la variable min, es decir, que el primer número es mi mínimo hasta ahora. Para el resto de los números, voy a hacer una comparación.** El valor de la variable bandera o flag puede ser, inicialmente cualquier valor y de cualquier tipo: 'a', 1, 0, etc. Lo que siempre tiene que hacer es cambiarle el valor una vez que hizo la tarea para la cual usó esa bandera.

```
Si bandera==1 entonces  min ← numero
                        bandera ← 0
```

c) En el ciclo mientras: La segunda vez que ejecuto las sentencias del ciclo y las veces siguientes:

sino Si numero < min entonces min \leftarrow numero

La segunda vez y el resto de las veces, el valor de la variable **bandera** es cero. Entonces la condición **bandera==1** es falsa. Entonces comparo si el número es menor al mínimo. Si es verdadero, lo asigno como mi nuevo mínimo. Si es mayor, no hago nada.

d) Análogamente se puede hacer la **búsqueda de un máximo**: Se asigna el primer valor como el máximo. El resto de los valores se compara contra el máximo. Si alguno de ellos es mayor que el máximo parcial, es el nuevo máximo.

4) Prueba de escritorio: Muy importante ejecutar el algoritmo con un juego de datos, para ver si realmente resuelve el problema:

Armamos una tabla con las variables del algoritmo, simulando lo que ocurre en la memoria de la computadora. Las variables del algoritmo van tomando valor si se ejecuta una instrucción de lectura o asignación. Se van ejecutando las instrucciones en el orden lógico desde el inicio al fin. Así queda la memoria al ejecutarse el algoritmo:

numero	bandera	min	contpi
	1		0
10	0	10	0
9	0	9	1
12	0	9	1
-4	0	-4	1
4	0	-4	2
0			

Muestra el cartel: **La cantidad de valores entre pi y 3pi es 2, el valor mínimo es -4.**

Probamos que nuestro algoritmo resuelve el problema.