

AEDD - Guía Práctica 4: Estructuras de Control Repetitivas o Iterativas

Introduciremos una técnica de uso frecuente utilizada para validar la resolución de problemas con algoritmos: **Pruebas de Escritorio**.

Una prueba de escritorio se utiliza para validar un algoritmo utilizando datos reales. El proceso para realizar una prueba de escritorio consiste en hacer seguimiento de un algoritmo recorriendo sus líneas secuencialmente, simulando el funcionamiento de la computadora. A medida que se van recorriendo las líneas se anotan en una tabla auxiliar los valores que van tomando las variables.

Para poder realizar una prueba de escritorio, es necesario como primera medida identificar cuáles son las variables de entrada, cuáles son las variables auxiliares y cuáles son las variables de salida. Una vez identificadas las variables, se debe distinguir el valor que toma cada una de ellas, a medida que se realizan las operaciones del algoritmo, utilizando para ello una tabla.

A continuación, se muestra un ejemplo sencillo para clarificar el concepto, suponiendo que tenemos el siguiente problema:

Escribir un programa que pida por teclado tres valores, a saber:

- la población actual de un país (*poblacionActual*),
- la tasa de crecimiento anual (*tasaCrecimiento*) y
- un valor de población esperada, al que desea llegar (*poblacionEsperada*).

Suponiendo que la tasa de crecimiento sea constante, determinar y mostrar por pantalla el año en el cual la población superará la población esperada.

Ejemplo:

Datos de entrada: 450000 60000 740000 donde <i>poblacionActual</i> : 450000 h <i>tasaCrecimiento</i> : 60000 h/a <i>poblacionEsperada</i> : 740000 h	Se informa por pantalla: En 5 años se alcanzará la población esperada.
---	--

Tabla de seguimiento de variables

	Variables				
Pasos	<i>poblacionActual</i>	<i>tasaCrecimiento</i>	<i>poblacionEsperada</i>	<i>contadorAnios</i>	Salida Standard

Se solicita codificar una solución en C++ para cada uno de los siguientes ejercicios utilizando el IDE ZinjaI. En todos los casos considerar que si los datos de entrada no son válidos, se debe mostrar un mensaje de error y finalizar la ejecución.

Ejercicios propuestos:

Tarea en línea URI: [Guía Repetitivas](#)

1. Escribe un programa que valide el ingreso de un número entero positivo. Si el número es negativo o cero, el programa debe solicitar el reingreso hasta que el número ingresado sea válido.
2. Escribir un programa que permita, dados dos valores enteros v1 y v2 que el usuario ingresa por teclado, mostrar por pantalla los números impares que existen en el rango [v1, v2], en orden descendente.

Ejemplo:

Se ingresan: 37 52

Los números impares entre 37 y 52 son: 51 49 47 45 43 41 39 37

3. Escribir un programa que lea por teclado un número entero K y dibuje en la pantalla una línea de K asteriscos.

Ejemplo: si K= 5:

* * * * *

4. Escribir un programa que lea por teclado un número entero K y dibuje en la pantalla un cuadrado cuyos lados estén representados por K asteriscos.

Ejemplo: si K= 5:

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

5. Escribir un programa que a partir del ingreso por teclado de un número entero K, dibuje un triángulo rectángulo, cuyos catetos estén representados por K asteriscos.

Ejemplo: si K= 5:

```
*
* *
* * *
* * * *
* * * * *
```

6. Realice un programa que encuentre y muestre todos los números de 4 cifras que cumplen la condición de que la suma de las cifras de orden impar es igual a la suma de las cifras de orden par.

Ejemplo de número que cumple con la condición: 3245 => 3 + 4 = 2 + 5

7. Un número es curioso si al ser multiplicado por sí mismo da como resultado otro número terminado en el número inicial dado. Por ejemplo, $6 \times 6 = 36$. Escribir un programa que solicite un número entero

positivo N y muestre los primeros N números curiosos.

8. Una empresa de ventas por correo vende cinco productos distintos, cuyos precios se muestran en la tabla siguiente:

Número de producto	Precio unitario
1	2.98
2	4.50
3	9.98
4	4.49
5	6.87

Escribir un programa que lea renglón a renglón una sucesión de pares de números, correspondientes a las ventas de una semana , como sigue:

Número-de-producto Cantidad-vendida-en-un-día

El programa deberá calcular y mostrar por pantalla:

- El producto que dio más ganancia por cada día.
- La ganancia total del día.
- El valor total facturado en la semana.

La carga termina cuando se ingresa el par 0 0

9. Dada una secuencia (no ordenada) de números enteros finalizada en CTRL+Z, determinar:
- Cantidad de números positivos.
 - Suma de los números positivos.
 - Promedio de números positivos.

Nota: El 0 (cero), al ser neutro, no se considera ni positivo ni negativo, por lo que no participará de las cuentas.

Ejemplo:

Se ingresan: 8, -3, 45, 9, -22, 3, 8, 79, -112, 0

Se informa:

Cantidad de positivos: 6

Suma de positivos: 152

Promedio de positivos: 25.33

10. Escribir un programa en lenguaje C++ que debe leer un conjunto de ternas con datos de un grupo de jugadores de fútbol: edad, posición (D: Defensor, M: Medio, A: Atacante), y goles anotados durante el último año. La secuencia de ternas finaliza con la terna (0,0,0).

Se debe informar:

- Porcentaje de jugadores mayores de 25 años.
- Cantidad de jugadores defensores que convirtieron más de 3 goles.
- Cantidad de goles convertidos por el goleador del grupo.
- Cantidad de atacantes que no convirtieron goles.
- Cantidad total de goles de los jugadores en los siguientes grupos etarios: 16-22, 23-29, 30-36.

Ejemplo:

27 D 2	Se informa: • Porc.de jugadores mayores de 25 años: 50%
22 M 4	
30 A 8	

25	A	12	• Cant. de defensores con más de 3 goles: 1
23	D	0	• Cant. de goles del goleador del grupo: 12
27	D	4	• Cant. de atacantes sin goles hechos: 0
18	D	0	• Cant. de goles según grupos etarios:
33	M	1	o 16-22 años: 4
			o 23-29 años: 18
			o 30-36 años: 9