

Práctico N° 4

Temas: Diseño: Composición Iterativa. Implementación. Prueba
Duración: 3 Clases

Esta práctica tiene como objetivos:

- Emplear las composiciones iterativas (repetir, mientras y para) en el diseño de algoritmos.
- Profundizar las nociones de condición, valores lógicos (verdadero y falso) y operadores lógicos (y, o, no).
- Resolver problemas que requieran la utilización de la composición iterativa (además de la secuencial y/o condicional).
- Realizar pruebas de escritorio sencillas en los algoritmos.
- Analizar ventajas y desventajas de utilizar cada una de las composiciones iterativas (repetir, mientras y para).
- Implementar en C algoritmos que contengan composición iterativa (además de la secuencial y/o condicional).
- Realizar pruebas sencillas en los programas.

Aclaración: En todos los ejercicios debes realizar el análisis del problema y entregarlo junto con cada algoritmo.

Ejercicios propuestos

1. Dados los algoritmos que presentamos a continuación:
 - a) Describe con tus palabras qué hace cada una de ellas.
 - b) Indica los posibles problemas que presentan.
 - c) ¿Cómo podrías solucionar cada uno de los problemas identificados?.

<p><u>Algoritmo</u> QueHago? (a)</p> <p><u>Léxico</u> msge ∈ Cadena</p> <p><u>Inicio</u> msge <-- "Hola"</p> <p><u>repetir</u> msge <-- "Estoy mareado %- " Salida: msge</p> <p><u>hasta que</u> falso msge <-- "Gracias!!!!" Salida: msge</p> <p><u>Fin</u></p>	<p><u>Algoritmo</u> Mafalda2 (b)</p> <p><u>Léxico</u> i ∈ Z msge ∈ Cadena</p> <p><u>Inicio</u> i ← 2 msge <-- "Mi dicho preferido es: " Salida: msge</p> <p><u>repetir</u> msge <-- "Paren el mundo " Salida: msge msge <-- "me quiero bajar." Salida: msge</p> <p><u>hasta que</u> i>=1 msge <-- "¿Les gustó?" Salida: msge</p> <p><u>Fin</u></p>
<p><u>Algoritmo</u> MostrarI (c)</p> <p><u>Léxico</u> i ∈ Z msge ∈ cadena</p> <p><u>Inicio</u> i ← 1</p> <p><u>repetir</u> msge ← "Soy i, mi valor es:" Salida: msge i i ← i + 1</p> <p><u>hasta que</u> i>=8 msge <-- "Soy yo de nuevo, valgo:" Salida: msge i</p> <p><u>Fin</u></p>	<p><u>Algoritmo</u> KtrasK (d)</p> <p><u>Léxico</u> k ∈ Z msge ∈ Cadena</p> <p><u>Inicio</u> msge <-- "Soy KTrasK" Salida: msge k <-- 2</p> <p><u>repetir</u> msge <-- "¿Cuánto valgo?" Salida: msge k k <-- k + 2</p> <p><u>hasta que</u> k=12 msge <-- "¿Les gustó?" Salida: msge</p> <p><u>Fin</u></p>

<p>Algoritmo MostrarI2 (e)</p> <p>Léxico</p> <p>$i \in \mathbb{Z}$</p> <p>$msge \in \text{cadena}$</p> <p>Inicio</p> <p>$i \leftarrow 1$</p> <p>si $i > 0$ entonces</p> <p> repetir</p> <p> $msge \leftarrow$ "Soy i, mi valor es:"</p> <p> Salida: $msge$ i</p> <p> $i \leftarrow i + 1$</p> <p> hasta que $i \geq 8$</p> <p> $msge \leftarrow$ "Soy yo de nuevo, valgo:"</p> <p> Salida: $msge$ i</p> <p> sino</p> <p> $msge \leftarrow$ "Solo se que no se nada"</p> <p> Salida: $msge$ i</p> <p> fsi</p> <p>Fin</p>	<p>Algoritmo KtrasK2 (f)</p> <p>Léxico</p> <p>$k \in \mathbb{Z}$</p> <p>$msge \in \text{Cadena}$</p> <p>Inicio</p> <p>$msge \leftarrow$ "Soy KTrasK2"</p> <p>Salida: $msge$</p> <p>$k \leftarrow 2$</p> <p>repetir</p> <p> si $k \leq 5$ entonces</p> <p> $msge \leftarrow$ "¿Soy chiquito?"</p> <p> Salida: $msge$ k</p> <p> sino</p> <p> $msge \leftarrow$ "¿Soy grande?"</p> <p> Salida: $msge$ k</p> <p> fsi</p> <p> $k \leftarrow k + 2$</p> <p> hasta que $k \geq 12$</p> <p> $msge \leftarrow$ "¿Les gustó?"</p> <p> Salida: $msge$</p> <p>Fin</p>
--	--

2. Desarrolla un Algoritmo que permita dar como resultado los números divisibles por 2, 4 y 6 que están comprendidos entre 1 y el número 100. Resuelve este algoritmo: Utilizando solo **repetir**.

3. Desarrolla un Algoritmo que realice la pregunta: ¿Desea continuar S/N? y que no deje de hacerla hasta que el usuario teclee N.

4. La sucesión de Fibonacci es famosa por que se encuentra en diversas formaciones naturales (en biología especialmente). Los primeros 10 números de la sucesión de Fibonacci son: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34... Los dos primeros de la serie son 0 y 1, a partir de estos los números se generan sumando los dos anteriores, así el tercero que es el 1 resulta de sumar $0 + 1$. Desarrolle un algoritmo que calcule e informe los n primeros números de la serie, siendo n un valor ingresado por el usuario, que es entero y positivo.

5. En una estación meteorológica, al final de cada año, necesitan saber cuál ha sido el mes en que hubo mayor precipitación y el mes en que hubo menor precipitación (lluvia caída). Los registros de lluvias se registran en forma mensual y se expresan en milímetros. Desarrolla un algoritmo que resuelva este problema. En caso de emplear composición iterativa, debe usar **repetir**.

6. Dados los algoritmos que presentamos a continuación:

- Describe con tus palabras qué hace cada uno de ellos.
- Indica los posibles problemas que presentan.
- ¿Cómo podrías solucionar cada uno de los problemas identificados?

<p>Algoritmo QueHago? (g)</p> <p>Léxico</p> <p>$msge \in \text{Cadena}$</p> <p>Inicio</p> <p>$msge \leftarrow$ "Hola"</p> <p>mientras Verdadero hacer</p> <p> $msge \leftarrow$ "Estoy mareado %- "</p> <p> Salida: $msge$</p> <p> fmientras</p> <p> $msge \leftarrow$ "Gracias!!!!"</p> <p> Salida: $msge$</p> <p>Fin</p>	<p>Algoritmo QueHago2? (h)</p> <p>Léxico</p> <p>$msge \in \text{Cadena}$</p> <p>Inicio</p> <p>$msge \leftarrow$ "Hola"</p> <p>mientras Falso hacer</p> <p> $msge \leftarrow$ "No soy de aquí!!!"</p> <p> Salida: $msge$</p> <p> fmientras</p> <p> $msge \leftarrow$ "Yo soy de allá!"</p> <p> Salida: $msge$</p> <p>Fin</p>
<p>Algoritmo Mafalda (i)</p> <p>Léxico</p> <p>$i \in \mathbb{Z}$</p> <p>$msge \in \text{Cadena}$</p> <p>Inicio</p> <p>$i \leftarrow 2$</p> <p>$msge \leftarrow$ "Mi dicho preferido es: "</p> <p>Salida: $msge$</p> <p>mientras $i < 4$ hacer</p> <p> $msge \leftarrow$ "Paren el mundo "</p> <p> Salida: $msge$</p> <p> $msge \leftarrow$ "me quiero bajar. "</p> <p> Salida: $msge$</p>	<p>Algoritmo MostrarI3 (j)</p> <p>Léxico</p> <p>$i \in \mathbb{Z}$</p> <p>$msge \in \text{cadena}$</p> <p>Inicio</p> <p>$i \leftarrow 1$</p> <p>mientras $i < 8$ hacer</p> <p> $msge \leftarrow$ "Soy i, mi valor es:"</p> <p> Salida: $msge$ i</p> <p> $i \leftarrow i + 1$</p> <p> fmientras</p> <p> $msge \leftarrow$ "Soy yo de nuevo, valgo:"</p> <p> Salida: $msge$ i</p> <p>Fin</p>

<pre> <u>fmientras</u> msge <-- "¿Les gustó?" Salida: msge <u>Fin</u> </pre>	
<pre> <u>Algoritmo</u> MostrarI4 (k) <u>Léxico</u> i ∈ Z msge ∈ cadena <u>Inicio</u> i ← 1 <u>si</u> i > 0 <u>entonces</u> <u>mientras</u> i < 8 <u>hacer</u> msge ← "Soy i, mi valor es:" Salida: msge i i ← i + 1 <u>fmientras</u> msge <-- "Soy yo de nuevo, valgo:" Salida: msge i <u>sino</u> msge <-- "Solo se que no se nada" Salida: msge i <u>fsi</u> <u>Fin</u> </pre>	<pre> <u>Algoritmo</u> KtrasK3 (l) <u>Léxico</u> k ∈ Z msge ∈ Cadena <u>Inicio</u> msge <-- "Soy KTrasK2" Salida: msge k <-- 2 <u>mientras</u> k < 12 <u>hacer</u> <u>si</u> k <= 5 <u>entonces</u> msge <-- "¿Soy chiquito?" Salida: msge k <u>sino</u> msge <-- "¿Soy grande?" Salida: msge k <u>fsi</u> k <-- k + 2 <u>fmientras</u> k >= 12 msge <-- "¿Les gustó?" Salida: msge <u>Fin</u> </pre>

I) PREGUNTA: Los ejemplos MostrarI (c) y MostrarI3 (j), ¿hacen lo mismo?

Justifique su respuesta explicando cómo se comportan las condiciones de ambas estructuras al comparar el valor de la variable **i**.

7. Se desea informar la suma de los números naturales entre 1 y un número determinado (n). Solucione el problema usando una estructura iterativa **mientras**. Por ejemplo, si se pasa el número 10 devuelve la suma de los números comprendidos entre 1 y 10: (1+2+3+4+...+10, el resultado es 55).

8. Un profesor de matemática necesita generar la tabla de multiplicar de un número entero comprendido entre 1 y 10. Por ejemplo para el 3 debería aparecer como salida:

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

.... y así hasta 10

Resuelva este problema utilizando un **mientras** y de modo que por la salida salga la tabla tal como se propone.

9. Un contador tiene que realizar el balance de un negocio. Su secretario ha preparado una lista de números que resumen el resultado de las facturas de todo el mes. En el listado aparecen números positivos (las ganancias) y negativos (las pérdidas). Puedes ayudar al contador sumando solo los valores que son negativos y descartando los positivos. El final de la lista se indica con un 999999. Usar un **mientras**.

10. Desarrollar un algoritmo que detecte si un número es primo o no. Un número es primo si sólo es divisible por sí mismo y por la unidad.

Ejemplo: 2, 3, 5, 7, 11, 17, 19 son números primos

9 no es número primo, es divisible por 1, 9, 3

Para resolver este problema hay que dividir sucesivamente el número estudiado por 2, 3, 5, etc., hasta el propio número.

Los números primos son enteros mayores que 1 sin divisores enteros positivos, exceptuando el 1 y ellos mismos. Todos los primos son impares, excepto el 2. Para saber si un número es primo sería suficiente comprobar la divisibilidad por números inferiores a la raíz cuadrada del número dado.

II) PREGUNTA: ¿Hay manera de reemplazar, un bloque de **repetir** por uno que utilice **mientras**? ¿Cómo?

III) PREGUNTA: ¿Hay manera de reemplazar, un bloque de **mientras** por uno que utilice **repetir**? ¿Cómo?

11. Desarrolla un Algoritmo que permita dar como resultado los números divisibles por 2, 4 y 6 que están comprendidos entre 1 y el número 100. Resuelve este algoritmo: Utilizando solo **para**.

12. En una escuela es necesario calcular el promedio de notas (las notas van de 0 a 10) de los alumnos de un curso, luego de tomar un determinado examen. Se necesita saber si el rendimiento ha sido elevado (el promedio es mayor a 8), aceptable (el promedio está comprendido entre 6 y 8) o bajo (promedio es inferior a

6). Desarrollar un algoritmo que resuelva este problema.

Para tener en cuenta: las autoridades del colegio saben cuántos estudiantes del curso han rendido el examen.

13. Desarrolle un algoritmo que permita calcular la potencia y de un número dado a , es decir: (a^y) . Tanto a como y son números enteros. El algoritmo debe poder ingresar los datos, calcular la potencia y dar como salida el resultado de: a^y .

14. Resolver el ejercicio 8 de esta práctica pero utilizando como estructura iterativa un **para**.

15. En los algoritmos que se dan a continuación, se deben reemplazar las estructuras iterativas que tiene el algoritmo por la estructura iterativa que se indica en cada punto:

a) Reemplazar **repetir** por un **mientras**.

Algoritmo CuentaRegresiva

Léxico

$k \in \mathbb{Z}$

$msge \in \text{Cadena}$

Inicio

$k \leftarrow 1$

repetir

$msge \leftarrow \text{"¿Salgo o no salgo?"}$

Salida: $msge$

$k \leftarrow k + 1$

hasta que $(k+5) > 20$

$msge \leftarrow \text{"Terminé!!!"}$

Salida: $msge$

Fin

b) Reemplazar **mientras** por un **para**

Algoritmo Impares

Léxico

$j \in \mathbb{Z}$

$msg \in \text{Cadena}$

Inicio

$j \leftarrow 1$

mientras $j \leq 20$ **hacer**

Salida: j

$j \leftarrow j + 2$

fmientras

$msge \leftarrow \text{"Terminé!!!"}$

Salida: $msge$

Fin

c) Reemplazar **para** por un **mientras**

Algoritmo SumatoriaEnteros

Léxico

$cotaSup \in \mathbb{Z}$ //cantidad de iteraciones del para

$i \in \mathbb{Z}$ //variable de control del para

$s \in \mathbb{Z}$ //sumatoria

$msg1, msg2 \in \text{Cadena}$

Inicio

Entrada: $cotaSup$

$s \leftarrow 0$

para ($i \leftarrow 0, i \leq cotaSup, i \leftarrow i + 1$) **hacer**

$s \leftarrow s + i$

fpara

$msg1 \leftarrow \text{"la suma de los "}$

$msg2 \leftarrow \text{"primeros números naturales"}$

Salida: $msg1$ $cotaSup$ $msg2$ s

Fin

d) Reemplazar **para** por un **mientras**
Algoritmo ContarHasta1

Lexico

$i \in \mathbb{Z}$

$msg \in \text{Cadena}$

Inicio

para ($i \leftarrow -10$, $i \leq 1$, $i \leftarrow i - 1$) **hacer**

$msge \leftarrow \text{"Hola, soy el"}$

 Salida: $msge$ i

fpara

$msge \leftarrow \text{"Terminé!!!"}$

 Salida: $msge$

Fin

Plan de Clases

Clase 1: 1), 4) y 5)

Clase 2: 6), 8) y 10)

Clase 3: 12) 13) y 15) Actividad: programa en C del 10)