



UNIVERSIDAD DE CARABOBO  
Facultad Experimental de Ciencias y Tecnología  
Departamento de Computación  
Unidad Académica de Algoritmos y Programación  
CS0218: Algoritmos y Programación II

## Taller #4

Año Lectivo 2009-2010

(Valor: 12 % del Componente Práctico)

### 1. Instrucciones

- Este taller de programación es una evaluación estrictamente individual; por lo tanto, durante la realización del mismo, no está permitido prestar o recibir ayuda de otro(s) estudiante(s).
- No está permitido el uso de dispositivos electrónicos o programables distintos al computador provisto para el taller.
- Pueden traer material bibliográfico (libros, guías, manuales, código impreso, etc.) para el desarrollo del taller de programación. También se permite el uso de material almacenado en medios magnéticos u ópticos.
- El estudiante que infrinja alguna de las instrucciones anteriores será retirado del taller y se le considerará aplazado con la nota mínima (Artículo 34 de las Normas de Evaluación de los Aprendizajes).
- **El desarrollo del taller tiene una duración estimada de 120 minutos.**
- El ejercicio propuesto requiere la lectura de datos de la entrada estándar (*standard input*) y la escritura de resultados en la salida estándar (*standard output*).
- Para el desarrollo del taller debe utilizar el lenguaje de programación C++, junto con sus librerías estándar.
- El ejercicio tiene asociado un nombre para el archivo que contiene el código fuente, el cual se indica posteriormente en el enunciado.

### 2. Evaluación de Árboles de Sintaxis

*Nombre del archivo con el código fuente: evaluar.cpp*

Una utilización común de los árboles binarios es la representación de expresiones aritméticas. Estas estructuras de datos se denominan árboles de sintaxis. Suponga la existencia de un árbol de sintaxis  $T$  que representa una expresión aritmética de la siguiente manera: los nodos hoja son cadenas que representan números enteros y los nodos interiores son cadenas que representan los operadores  $^*$ ,  $+$ ,  $-$ ,  $/$ . Diseñar una función “evaluar” que reciba como parámetro un árbol de sintaxis con estas características y retorne el resultado de la evaluación de la expresión representada.

#### 2.1. Formato de Entrada

La entrada consistirá de una serie de pares de líneas que contendrán los recorridos en (preorden, inorden) o (postorden, inorden), para la reconstrucción de los árboles de sintaxis. Cada línea tendrá dos valores: el primer valor indica el tipo de recorrido (PREORDEN, INORDEN, POSTORDEN) y el segundo es la expresión matemática.

*La entrada debe leerse desde la entrada estándar (standard input).*

## 2.2. Formato de Salida

La salida mostrará los resultados de cada uno de los casos de prueba, según el formato que se muestra en el ejemplo.

*La salida deberá escribirse en la salida estándar (standard output).*

## 2.3. Ejemplo de Entrada

```
1 PREORDEN + - * 5 7 3 * 6 + 2 1
2 INORDEN 5 * 7 - 3 + 6 * 2 + 1
3
4 POSTORDEN 5 7 * 3 - 6 2 1 + * +
5 INORDEN 5 * 7 - 3 + 6 * 2 + 1
6
7 PREORDEN * + 1 / 2 3 - 4 * 5 6
8 INORDEN 1 + 2 / 3 * 4 - 5 * 6
9
10 POSTORDEN 1 2 3 / + 4 5 6 * - *
11 INORDEN 1 + 2 / 3 * 4 - 5 * 6
12
13 PREORDEN - + 2 3 / 8 + 2 3
14 INORDEN 2 + 3 - 8 / 2 + 3
15
16 POSTORDEN 2 3 + 8 2 3 + / -
17 INORDEN 2 + 3 - 8 / 2 + 3
```

## 2.4. Salida para el Ejemplo de Entrada

```
1 Caso #1:
2 50
3
4 Caso #2:
5 50
6
7 Caso #3:
8 -43.33
9
10 Caso #4:
11 -43.33
12
13 Caso #5:
14 3.4
15
16 Caso #6:
17 3.4
```