



UNIVERSIDAD DE CARABOBO
Facultad Experimental de Ciencias y Tecnología
Departamento de Computación
Unidad Académica de Algoritmos y Programación
MCPII-2016: Programación II

Taller #2

Año Lectivo 2016

(Valor: 30 % 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 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.

2. Representación sin Fronteras

Se define por *frontera* de un árbol binario, la secuencia formada por los elementos almacenados en las hojas de un árbol binario, tomados de izquierda a derecha.

Desarrolle el método *eliminarFrontera* dentro de su clase Árbol Binario de manera tal que, dado un árbol binario A , se genere un nuevo árbol binario B y que éste sea la representación sin frontera de A .

2.1. Formato de Entrada

La primera línea de la entrada contiene un número entero k ($0 < k \leq 100$), que representa el número de casos de prueba. Cada uno de los k casos consiste de tres líneas. La primera línea contiene un número entero n ($0 < n \leq 1000$) que representa el número de elementos del árbol. En la segunda línea hay n números enteros, separados entre sí por un espacio en blanco, que representan el recorrido del árbol en preorden en caso de que k sea impar y en postorden en caso de que k sea par. En la tercera línea hay n números enteros, separados entre sí por un espacio en blanco, que representan el recorrido en inorden.

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

2.2. Formato de Salida

Para cada caso k en la entrada, escriba primero una línea con el siguiente formato: “Caso # k :”. En la línea siguiente, escriba el recorrido del árbol binario generado B en preorden en caso de que k sea impar o en postorden en caso de que k sea par. En la siguiente línea, escriba el recorrido en inorden del árbol binario generado B . Coloque una línea en blanco después de cada caso de prueba.

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

2.3. Ejemplo de Entrada

```
1 2
2 9
3 8 6 3 2 7 12 10 13 14
4 2 3 6 7 8 10 12 13 14
5 9
6 2 3 7 6 10 14 13 12 8
7 2 3 6 7 8 10 12 13 14
```

2.4. Salida para el Ejemplo de Entrada

```
1 Caso #1:
2 8 6 3 12 13
3 3 6 8 12 13
4
5 Caso #2:
6 3 6 13 12 8
7 3 6 8 12 13
8
```

Observaciones

- Los programas que entreguen deben estar debidamente presentados y documentados con el nombre del autor, cédula de identidad y sección a la que pertenece.
- Los archivos deben ser enviados a través del entorno virtual de la asignatura.
- Los archivos deben ser colocados en un directorio y luego comprimirlos. Sólo debe enviar el código fuente de sus programas y el *Makefile*. Su taller no será corregido si no envía el *Makefile* junto con su código fuente. Asegúrese de enviar todos los archivos que forman parte de su solución.
- **EL INCUMPLIMIENTO DE ESTAS OBSERVACIONES CONLLEVARÁ A LA NO REVISIÓN DE SU TALLER.**