Universidad de Carabobo
Facultad Experimental de Ciencias y Tecnología
Departamento de Computación
CS0218 Algoritmos y Programación II

Taller Nº 3: Estructuras Jerárquicas

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).
- **ESTÁ** permitido el uso de códigos fuentes en digital hechos por usted para el desarrollo del taller.
- Pueden traer material bibliográfico (libros, guías, manuales, código impreso, etc.) para el desarrollo del taller de programación.
- 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 tres (03) horas académicas.
- 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. Para la compilación de sus códigos fuentes, debe realizarla por medio de un archivo ***makefile***, el cual deberá entregar junto con sus códigos fuentes.

2. OBJETIVO

Se espera que, al finalizar esta actividad, el estudiante esté en capacidad de desarrollar estructuras dinámicas mediante la utilización de apuntadores, para la posterior resolución de problemas que requieren el manejo de Estructuras Jerárquicas.

3. ACTIVIDADES

Un árbol binario b_1 es ***menor*** que otro b_2, si todos los elementos de b_1 son menores que todos los elementos de b_2. Implemente un algoritmo que dado dos arboles binarios de enteros b_1 y b_2 determine si b_1 es menor a b_2.

4. Formatos de Entrada y Salida

Entrada

La entrada constará de varios casos de prueba, cada caso de prueba está conformado por dos pares de líneas, que contendrán los recorridos en (preorden, inorden) o (postorden, inorden), para la construcción de árboles binarios. Cada línea comenzará con una palabra indicando el tipo de recorrido (PREORDEN, INORDEN, POSTORDEN), seguida de un entero ***n*** separado por un espacio en blanco, indicando el numero de elementos en el recorrido, seguido por ***n*** enteros (no repetidos) separados por un espacio en blanco, que constituyen el recorrido. El primer par de líneas corresponden a los recorridos del árbol b_1 y el segundo par corresponden a los recorridos del árbol b_2.

Salida

La salida constará de una línea por cada caso de prueba indicando si el árbol b_1 es menor o no el árbol b_2.

Ejemplo

Entrada

```
PREORDEN  3 1 2 3
INORDEN 3 2 1 3

POSTORDEN 3 5 6 4
INORDEN   3 5 4 6

POSTORDEN 3 50 30 20
INORDEN   3 50 20 30

PREORDEN  2 40 19
INORDEN   2 19 40
```

Salida

```
Es menor
No es menor
```

5. Observaciones

- No debe mostrar por pantalla ningún tipo de mensaje al usuario para su interacción con el programa, ninguna interfaz gráfica desarrollada por usted, será corregida positivamente o le aportara ningún punto extra para su calificación final
- El formato de salida debe ser respetado como se indica en el enunciado.
- Debe proveer un archivo ***makefile*** que permita compilar su taller, la falta del mismo conllevará a la no revisión de su taller.