UNIVERSIDAD DE CARABOBO
Facultad Experimental de Ciencias y Tecnología
Departamento de Computación
U.A. de Algoritmos, Programación y Lenguajes
CAO403: Programación II

Taller #2

Período Lectivo 1-2016
(Valor: 30%)

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).
- ESTA 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.
- Debe proveer un archivo ***makefile*** que permita compilar su taller, la falta del mismo conllevará a la no revisión de su taller, **adicionalmente dicho *makefile* debe generar un ejecutable de nombre** "**binario",** con el fin de facilitar las labores de corrección de su taller, el no hacerlo generara puntos menos en su nota final del taller.

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

Se dice que un **árbol binario es de búsqueda** si todos los elementos del subárbol izquierdo son menores que la raíz, la raíz es menor que todos los elementos del subárbol derecho y los dos subárboles asociados son también árboles binarios de búsqueda. Por definición, el árbol vacío es un árbol binario de búsqueda. Implemente un algoritmo recursivo que dado un (1) **árbol binario *A*** de **enteros** determine si ***A*** es **binario de búsqueda**. Su solución debe ser a lo sumo de complejidad ***O(n).*** **UTILICE SÓLO APUNTADORES PARA SU SOLUCIÓN. NO PUEDE UTILIZAR ESTRUCTURAS AUXILIARES DE NINGÚN TIPO. NO PUEDE REALIZAR EL PROCESAMIENTO DE DATOS DIRECTAMENTE SOBRE LA ENTRADA.**

4. Formatos de Entrada y Salida

Entrada

La entrada constará de varios casos de prueba, cada caso de prueba está conformado por 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.

Salida

La salida constará de una línea por cada caso de prueba indicando si el árbol binario *A* es o no de búsqueda

Ejemplo

Entrada

```
PREORDEN 7 4 2 1 3 6 5 7
INORDEN 7 1 2 3 4 5 6 7
POSTORDEN 7 1 8 2 5 7 6 4
INORDEN 7 1 2 8 4 5 6 7
```

Salida

```
ES binario de busqueda.
NO es binario de busqueda.
```

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, de lo contrario acarreará con puntos menos en su nota final.