

Tarea 3

Programación en C: TDAs recursivos – Árboles

Fecha de envío: Domingo 26 de Junio, a las 23:55h
Modalidad: Trabajo en grupos de a lo más **dos** personas

1. Objetivos

El objetivo de este Laboratorio es evaluar su capacidad para:

- Revolver un problema que requiera búsqueda y ordenamiento de información aplicando árboles.

2. Enunciado

En este laboratorio utilizaremos el mismo contexto que en el laboratorio 1, pero en esta ocasión deberás utilizar un árbol binario ordenado para determinar las personas que están intentando comprar más de las dos entradas que se permitían en la promoción. Para lograr lo anterior, tu árbol debe tener como clave el rut de la persona. En esta ocasión, los ruts no tendrán el dígito verificador por lo que puedes usar números enteros para representarlos. Adicionalmente, en cada uno de los nodos del árbol deberás almacenar la cantidad de entradas que la persona desea comprar. Debes registrar una única vez en el árbol cada uno de los ruts que estén en el archivo de entrada. La primera vez que se crea el nodo, el número de entradas es el que aparece en la línea del archivo con que se creó el registro. Si una persona aparece más de una vez en el archivo de entrada, la segunda vez debes buscar su registro en el árbol usando el rut como clave y sumarle la cantidad de entradas adicionales que la persona está intentando comprar. Como resultado final tu programa debe generar un archivo que indique las personas que intentaron adquirir más de las dos entradas permitidas en la promoción indicando la cantidad total solicitada.

El formato de entrada corresponde a los ejemplos mostrados en la tabla 1.

2.1. Lo que se pide en este laboratorio

Al ejecutar tu programa, éste debe solicitar el nombre del archivo de entrada y de salida al usuario. Debe permitir que el usuario ingrese la extensión que desea (ver figura 1)

Representación en el archivo de texto
12333444,Maria Reyes,2
13444555,Pedro Diaz,1
19000111,José Duran,1
13444555,Pedro Diaz,2

Tabla 1: ejemplos del formato en el archivo de entrada datos.txt

```

elizabeth@cancun: $ ./a.out
Ingrese el nombre del archivo: _
Ingrese el nombre del archivo de salida: _

```

Figura 1: Mensaje inicial al usuario al ejecutar su programa.

```

12333444,Maria Reyes,2
13444555,Pedro Diaz,1
19000111,José Duran,1
13444555,Pedro Diaz,2

```

Figura 2: ejemplo de contenido de un archivo de prueba usado por el usuario.

Suponiendo que el archivo que ingresa el usuario tiene el contenido mostrado en la figura 2. En la figura 3 aparece la información que debiera aparecer en el archivo de salida.

```

13444555 3

```

Figura 3: Ejemplo de la salida que debe tener su programa al procesar el archivo ingresado por el usuario de la figura 2.

3. Consideraciones en la Revisión

3.1. Sobre el código

- Tu programa debe seguir exactamente las reglas e interfaces explicadas en este documento.
- Debes construir **una función para cada tarea que realice tu código**, cuidando los tipos de datos de las entradas y de las salidas.
- Debes usar identificadores representativos para tus variables, parámetros de entrada y funciones.
- Debes comentar cada una de tus funciones, estructuras y tipos de datos que definas, indicando una descripción de la labor que lleva a cabo, sus entradas, y sus salidas.

- Tu código debe estar correctamente indentado (uso de sangrías para cada sub-bloque de instrucciones), esto incluye el correcto alineamiento de las llaves ({ y }) que enmarcan tales bloques. Se penalizarán códigos desordenados en este aspecto.
- Tu código no puede presentar más de 1 línea en blanco. Se penalizarán códigos que no cumplan con este punto.
- Puedes trabajar con el IDE y compilador de C, que más te acomode. No obstante lo anterior, tu código debiera poder ser compilado y ejecutado sin problemas en Windows.
- El sistema debe ser robusto, se penalizarán los errores no manejados, de cualquier tipo.

4. Sobre la entrega y faltas a la ética

1. Debes subir tu trabajo al aula virtual de tu curso de cátedra dentro de la plataforma <https://unab.blackboard.com/>, en una casilla que se habilitará especialmente para esto.
2. **NO SE ACEPTARÁN TRABAJOS ATRASADOS.**
3. Tu trabajo debe ser subido por **SOLO UNO de los integrantes del grupo.**
4. Debe subir un archivo comprimido (.rar o .zip) que incluya los siguientes archivos:
 - Todos los códigos fuentes (archivos .h y .c).

Todos estos archivos deben ser comprimidos en un solo archivo llamado “rutcompleto1_rutcompleto2.zip”, indicando el rut de cada integrante del grupo. Por ejemplo, si tu rut es 19.000.111-3, debieras ingresarlo sin puntos ni guión, pero sí con el dígito verificador: o sea 190001113.

En caso de que el archivo presente errores al descomprimirse la nota será **1.0**.

5. Ante la ausencia de alguno de los archivos mencionados en el punto anterior, la nota del trabajo completo será un **1.0**.
6. Si el programa no se puede ejecutar, tendrá la nota mínima: **1.0**. Si su programa funciona, se evaluará su ejecución, y también el código fuente.
7. Ante el escenario de existir sospecha de copia (con otros compañeros, o desde internet), será interrogado acerca de su trabajo, para aclarar dudas de su entendimiento y autoría. Si se confirma la sospecha, el trabajo será evaluado con nota **1.0**.
8. Las consultas las debe realizar directamente a los profesores de taller, presencialmente en clases o a los correos y en los horarios que ellos establezcan.