Estructuras de Datos - Abril 30, 2024

Taller 2: Instrucciones

Prof. Mauro San Martín

Este documento define los requisitos y da instrucciones para la realización del *Taller 2* de la asignatura de Estructuras de Datos. El *Taller 2* es una actividad grupal con evaluación individual. Es responsabilidad de cada estudiante sistematizar la evidencia de su trabajo y aportes al proyecto grupal. La plataforma recomendada para apoyar la colaboración y dejar un registro del trabajo realizado por cada integrante del grupo es *GitHub*.

Definición del problema y requisitos de la solución. Existen problemas que requieren representar números enteros más allá de la capacidad de los tipos de C. Por ejemplo, como se pudo constatar en la actividad previa para el cálculo del factorial ($n!$). En lugar de implementar soluciones puntuales para operaciones específicas, se desea implementar el tipo numérico `EnteroLargo` que permita representar números enteros muy grandes, por ejemplo con cientos o miles de dígitos, y realizar sobre variables de este tipo las cuatro operaciones aritméticas básicas: suma ($+$), resta ($-$), multiplicación ($*$), y división ($/$). Estas operaciones básicas deberían servir para implementar operaciones más complejas, como por ejemplo el mismo factorial.

Los requisitos que debe cumplir este nuevo tipo de datos y su interfaz son los siguientes:

1. (10 %) El tipo de datos debe ser autocontenido. Cada variable declarada de este tipo de datos debe contener, al menos: todos los dígitos decimales del número, su signo, y su cantidad de dígitos. Además el tipo de datos debe considerar el uso eficiente de la memoria, y almacenar el número en una lista enlazada, donde cada nodo contenga 1 dígito.

2. (30 %)Deben existir al menos las funciones para crear, asignar, eliminar y archivar variables de este tipo, ejemplificadas a continuación:

   - `EnteroLargo *creaEnteroLargoDesdeStr("123");` crea y retorna una referencia a un `EnteroLargo` inicializado con el valor representado por el string dado, en este caso 123.
   - `char *creaStrDesdeEnteroLargo(numero);` retorna una referencia a un string conteniendo el valor actualmente almacenado en la variable de tipo `EnteroLargo` referenciada por `numero`. Por ejemplo, si

`numero` fue inicializada según el ejemplo previo, esta función debería retornar el string "123".

- `int copiaEnteroLargo(destino, fuente);` copia el número en la variable `fuente` a la variable `destino` (ambas de tipo `EnteroLargo`), equivale a la asignación `destino = fuente`.
- `int eliminaEnteroLargo(numero);` libera la memoria ocupada por un `EnteroLargo` referenciado por la variable `numero`.
- `EnteroLargo *leeEnteroLargo(nombreArchivo);` lee del archivo identificado por `nombreArchivo` y retorna una referencia a un nuevo `EnteroLargo` con el valor leído.
- `int escribeEnteroLargo(nombreArchivo, numero);` escribe el `EnteroLargo` referenciado por `numero` en el archivo llamado `nombreArchivo`.

3. (30 %)Deben existir funciones que implementen la suma y la resta:

   - `int igualEnteroLargo(numero1, numero2);` retorna 0 si los números referenciados por `numero1` y `numero2` (ambos de tipo `EnteroLargo`) son iguales.
   - `EnteroLargo *sumaEnteroLargo(numero1, numero2);` retorna una referencia a un nuevo `EnteroLargo` con el valor de la suma de los números referenciados por `numero1` y `numero2` (ambos de tipo `EnteroLargo`).
   - `EnteroLargo *restaEnteroLargo(numero1, numero2);` retorna una referencia a un nuevo `EnteroLargo` con el valor de la resta de los números referenciados por `numero1` y `numero2` (ambos de tipo `EnteroLargo`).

4. (30 %) Deben existir funciones para multiplicar y dividir:

   - `EnteroLargo *numero = multiplicaEnteroLargo(multiplicando, multiplicador);` retorna una referencia a un nuevo `EnteroLargo` con el valor de la multiplicación de los números referenciados por `multiplicando` y `multiplicador` (ambos de tipo `EnteroLargo`).
   - `EnteroLargo *numero = divideEnteroLargo(dividendo, divisor);` retorna una referencia a un nuevo `EnteroLargo` con el valor de la división entera de los números referenciados por `dividendo` y `divisor` (ambos de tipo `EnteroLargo`).

Instrucciones.

1. Cada grupo debe implementar el tipo `EnteroLargo` según los requisitos dados, y proporcionar una función `EnteroLargo` que muestre su funcionamiento.

2. Los ejemplos dados de las funciones a implementar definen la interfaz de cada función, defina las funciones en su proyecto de acuerdo a ellos, no introduzca cambios. Esto es indispensable para que funcione el código de prueba y evaluación.

3. Cada tipo de datos y sus funciones debe estar implementado en un archivo aparte (`tipo.h`). Vea la plantilla del proyecto.

4. Cada grupo debe mantener actualizado el archivo informativo del repositorio con las características de su implementación, y cualquier información relevante al momento de utilizarla.

5. Cada integrante del grupo debe tener tareas que desarrollar en todo momento. Organice su trabajo y actualice el repositorio del proyecto regularmente con sus aportes. La evaluación se hará durante todo el proyecto, monitoreando regularmente los avances grupales e individuales.
   IMPORTANTE: Para la evaluación son igualmente importantes el trabajo y registro sistemático del progreso, como lograr implementar correctamente los requisitos del taller.

6. Para la evaluación de los objetivos alcanzados en el taller se considerará como versión final la disponible en el repositorio el **miércoles** 05 **de junio a las 14:00 horas**.

Evaluación.

1. Cada grupo de requisitos tiene la ponderación indicada en su descripción (totalizando 100 %).

2. Cada requisito será calificado entre 0 y 6,0. Para que se considere logrado su implementación debe ser correcta (50 %), y el código debe estar bien organizado (25 %) y ser legible (25 %).

3. La calificación grupal G será la suma de las calificaciones de los requisitos del taller según las ponderaciones indicadas.

4. El aporte individual I se medirá con un valor entre 0 y 1,0 (o 100 %), reflejando si el esfuerzo y aporte al resultado del taller del o la integrante, documentado en el registro sistemático de su trabajo o determinado mediante una interrogación, es adecuado respecto al esfuerzo esperado dada la cantidad de integrantes del grupo. Por ejemplo, si el grupo tiene cuatro integrantes se esperará que el esfuerzo se divida uniformemente en cuatro (25 % cada integrante), entonces si un integrante demuestra un esfuerzo $\geq$ 25 % su aporte individual será de 1,0. En ningún caso un integrante podrá obtener un $I > 1{,}0$. La calificación del taller T para cada integrante corresponderá entonces a $G * I$.