PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 1

8va práctica (tipo b)
(Segundo Semestre 2019)

Indicaciones Generales:

- Duración: 110 minutos.
- Se podrá usar como material de consulta solo sus apuntes de clase (NO fotocopias, impresos ni hojas sueltas).
- No se pueden emplear **variables globales**, **ni estructuras**. Tampoco se podrán emplear la **clase string** ni las funciones malloc, realloc, strdup o strtok, igualmente no se puede emplear cualquier función contenida en las bibliotecas stdio.h, cstdio o similares y que puedan estar también definidas en otras bibliotecas.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada módulo **NO** debe sobrepasar las 20 líneas de código aproximadamente. El archivo **main.cpp** solo podrá contener la función **main** de cada proyecto y el código contenido en él solo podrá estar conformado por invocaciones a los métodos de la clases. En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le **descontará 0.5 puntos por archivo**. Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- NO SE CALIFICARÁN AQUELLAS FUNCIONES DESARROLLADAS EN EL MISMO ARCHIVO QUE LA FUNCIÓN main.
- El código comentado NO SE CALIFICARÁ.
- **NO PODRÁ DEFINIR FUNCIONES QUE NO ESTÉN LIGADAS A UNA CLASE.**
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- **La presentación, la ortografía y la gramática de los trabajos influirá en la calificación.**

Puntaje total: 20 puntos

Problema

La mayoría de lenguajes de programación presenta limitaciones con respecto al manejo de valores numéricos enteros. Esto es, los tipos de dato que se manejan están definidos por lo general en 1, 2, 4, 8,... bytes, lo que restringe el valor máximo que podemos emplear en un programa.

Un problema que se presenta con frecuencia al implementar aplicaciones científicas es que muchas veces en estas aplicaciones se requieren manipular números muy grandes, estamos hablando de valores con 20, 30 o más cifras.

Este laboratorio pretende que usted defina una clase que a través de sus objetos permita definir, manipular y operar números muy grandes.

DEBERÁ LEER Y ENTENDER TODAS LAS TAREAS SOLICITADAS ANTES DE EMPEZAR A ESCRIBIR EL CÓDIGO DE PROYECTO PARA EVITAR REPETIR CÓDIGO INNECESARIAMENTE.

PARTE 1: Clase NumeroMuyGrande (15 puntos)

Se solicita que desarrolle un proyecto, denominado **LAB08_PREG01_NumeroMuyGrande**, en el cual se defina una clase denominada "**NumeroMuyGrande**" (en archivos **NumeroMuyGrande.cpp** y **NumeroMuyGrande.h**). La clase **NumeroMuyGrande** definirá los atributos y métodos (datos y funciones miembro) necesarios que cumplan con los siguientes requerimientos:

(1) Atributos (1 punto):

El **número** se manejará de manera dinámica a través de un puntero `char*` (recuerde que el tipo char define un entero de un byte de tamaño, en el rango de -128 a 127). Asimismo, deberá almacenar la **cantidad de cifras** del número (**int**), de modo que no se tenga que contar las cifras cada vez que se le

solicite y que se actualice en cada operación. Finalmente el **signo** (**char**) del número: '**+**' o '**-**'. De acuerdo a esto, la clase estará definida únicamente por los atributos: ***numero, cantCifras y signo***.

(2) <u>Inicialización</u> (3.0 puntos):

Se podrá inicializar un objeto de esta clase de las formas siguientes:

- ✓ **class NumeroMuyGrande num01;** El contenedor se inicializa en NULL, la cantidad de cifras será cero y el signo será '+'.
- ✓ **class NumeroMuyGrande num02("547382072206391");** La cadena recibida es procesada de modo tal que **<u>cada carácter será convertido a su valor numérico</u>** y esta cifra almacenada en el arreglo contenedor, este arreglo se definirá con una longitud exacta y solo contendrá las cifras del número. Para poder realizar operaciones con estos números de manera más fácil, en número será almacenado de la misma manera como lo hace el computador con sus datos, esto es, al revés de la forma como una persona los lee y entiende. Para el caso del ejemplo:

  "547382072206391" ⇒

1	9	3	6	…	3	7	4	5

  Tenga en cuenta que'5' ≠ 5
  Y lo que se debe guardar es 5

  El signo dependerá del parámetro: **"+54…" o "54…" ⇒ '+', "-54…" ⇒ '-'**

- ✓ **class NumeroMuyGrande num02 = num01;** este constructor copiará de manera idéntica, toda la información de **num01** en num**02**. Los punteros <u>NO</u> deben apuntar al mismo dato.

(3) <u>Asignación</u> (4 puntos):

A un objeto de clase **NumeroMuyGrande** se le podrá asignar valores de las formas siguientes:

- ✓ **Num01.asignar("73468428428642");** Asigna el argumento al objeto, de la misma manera como lo hace el constructor.
- ✓ **Num01.asignar(num02);** Hace lo mismo que el método anterior pero la asignación se hace desde un objeto tipo class **NumeroMuyGrande** enviado como parámetro.
- ✓ **num01 = "123456789012345";** Igual a lo anterior pero sobrecargando **operador =**
- ✓ **num01 = num02;** Sobrecargando **operador =** a partir de un objeto tipo class **NumeroMuyGrande**.

(4) <u>Operaciones</u> (3 puntos):

- ✓ **num03 = num01 + num02;** realiza una **suma aritmética** entre los números contenidos en los objetos **num01** y **num02**. El resultado es asignado a otro objeto del mismo tipo. Los espacios de memoria serán exactos.
- ✓ **num03 = num01 - num02;** realiza una **resta aritmética** entre los números contenidos en los objetos **num01** y **num02**. El resultado es asignado a otro objeto del mismo tipo. Los espacios de memoria serán exactos.

(5) <u>Comparación</u> (2 puntos):

Se podrá comparar dos objetos de esta clase de las formas siguientes:

- ✓ **num01.compare(num02);** → Compara el objeto ingresado como parámetro con la del objeto **num01**. Devuelve cero si son iguales, un valor mayor que cero si el número del objeto es mayor que el del parámetro y un valor menor que cero si es menor.
- ✓ **num01 == num02**, **num01 > num02** y **num01 < num02** → Se comparan dos objetos tipo class **NumeroMuyGrande** mediante la sobrecarga de los operadores. Se devuelve **true** si se cumple la condición y **flase** si no. Estas sobrecargas deben implementarse como métodos de la misma clase (no como una función externa).

(6) <u>Otros métodos</u> (2 puntos):

- ✓ **<u>Longitud</u>** en la forma: **num01.length()** → Devuelve la cantidad de cifras del número (no se debe contar cifras aquí).
- ✓ **<u>Lectura</u>** Lee **<u>de un archivo de textos</u>** el número, para esto debe sobrecargar el **operador >>** para **ifstream**.

- ✓ **Impresión** Imprime en un archivo el número **tabulado y alineado a la derecha**, para esto debe sobrecargar el **operador <<** para **ofstream**.
- ✓ **Destructor** Debe liberar los espacios de memoria dinámica asignados al objeto.

(*) Pruebas individuales:

Deberá elaborar un conjunto de pruebas (en el archivo **main.cpp** del proyecto) que permita probar de manera sencilla cada uno de los métodos y sobrecargas. Los métodos u operadores sobrecargados que no sean parte de esta prueba no serán calificados.

PARTE 2: Prueba final (5 puntos)

Se solicita que desarrolle un proyecto denominado **LAB08_PREG01_NumeroMuyGrande** en el cual utilizará la clase **NumeroMuyGrande** implementada en la primera parte (copie los archivos **NumeroMuyGrande.cpp** y **NumeroMuyGrande.h** de su anterior proyecto).

Empleando un archivo de textos como el que se muestra a continuación:

```
6446826237682   -83475838234   +1289182009348392
645737   812121294
…
```

En cada línea puede haber más de un número muy grande

Usted deberá leer el archivo y guardar los números en un **arreglo estático de tipo NumeroMuyGrande**. Luego, desde dicho arreglo, emitir un reporte en un archivo de textos, en el que se muestre la suma de todos los números grandes, la resta de todos los números grandes, la cantidad de datos iguales, mayores o menores que el primero. Finalmente, debe imprimir el listado de números (uno por línea) tabulados y alineados a la derecha, los números deben poder ser leídos y entendidos por cualquier persona, por lo que no se tomara en cuenta el reporte si los imprime al revés.

Anotaciones finales

Al finalizar el laboratorio, comprima[1] la carpeta "Laboratorio8" en un archivo con nombre <código del alumno con 8 dígitos>.zip y súbalo a la intranet del curso, en el enlace Documentos, en la carpeta \Laboratorio8\<código del horario>\<aula>.

Profesores del curso: Rony Cueva
Miguel Guanira E.

San Miguel, 15 de diciembre del 2019.

[1] Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip) no 7z.