

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 1

2da práctica (tipo b)
(Segundo Semestre 2013)

Indicaciones Generales:

Está permitido usar todo tipo de material de consulta.

No se permite el uso de estructuras, clases, objetos, ni la función fopen. Tampoco se permite el uso de las bibliotecas string.h, cstring, string o similares, ni el formato %s. Sólo se usarán arreglos estáticos. No se podrá trabajar con punteros dinámicos, esto es no podrán emplear malloc, realloc, new, etc.

Duración: 1 hora 50 minutos.

Puntaje total: 20 puntos.

Objetivo del laboratorio: Que el alumno construya una "Biblioteca estática" y la use en un proyecto.

Problema

Muchas veces cuando trabajamos con números, sobre todo cuando hacemos cambios de base, nos hemos dado con la sorpresa que se producen desbordamientos, esto es que el valor final sobrepasa los límites permitidos para almacenar el valor en la variable. Este problema se presenta también cuando las cantidades que se manejan en un programa son muy grandes.

Se desea elaborar una biblioteca estática de funciones en C/C++ que permita manipular números en diferente base pero puedan tener una cantidad grande de cifras. La biblioteca manipulará los números mediante arreglos, en los que en cada elemento del arreglo se guardará una cifra del número. Todos los arreglos que manipularán los números serán del tipo entero char y las cifras se guardarán como número mas no como caracter. Las bases con las que trabajará el programa estarán sólo entre 2 y 10. La biblioteca constará de una serie de funciones, cuyos prototipos se muestran a continuación:

a) **int lee(int base, char numG[]):**

Lee del medio estándar de entrada una secuencia de cifras, sin separadores entre las cifras y las colocará en el arreglo. Si el número tiene separadores al inicio los ignorará. Si durante la lectura alguna de las cifras no corresponde con la base, la función devolverá cero. Si se intenta leer el número y se encuentra el final del archivo antes de haber leído la primera cifra, se devuelve EOF. En cualquier otro caso se devuelve un valor diferente de EOF o cero (decídalo usted).

La función, antes de retornar, deberá colocar las cifras significativas del número en el extremo derecho del arreglo.

El número ingresado podrá tener signo. Quedará a su criterio la forma cómo lo maneja en esta función y en las siguientes.

b) **int suma(int base, char numG1[], char numG2[], char resp[]):**

Esta función suma dos números grandes que se encuentran en una base dada, y coloca la respuesta en otro arreglo. Si el número se desborda se devolverá 1 y en otro caso cero.

La operación se hará directamente en la base del número, no se permitirá convertir los números a otra base para realizar la operación. Debe respetar los signos de los números.

c) **int resta(int base, char numG1[], char numG2[], char resp[]):**

Esta función resta dos números grandes que se encuentran en una base dada, y coloca la respuesta en otro arreglo. Si el número se desborda se devolverá 1 y en otro caso cero.

La operación se hará directamente en la base del número, no se permitirá convertir los números a otra base para realizar la operación. Debe respetar los signos de los números.

d) **int multCifra(int base, char numG1[], char cifra, char resp[]):**

Esta función multiplica un número grande que se encuentran en una base dada con un número de una cifra, y coloca la respuesta en otro arreglo. Si el número se desborda se devolverá 1 y en otro caso cero.

La operación se hará directamente en la base del número, no se permitirá convertir los números a otra base para realizar la operación. Debe respetar los signos de los números.

e) `int multiplica(int base, char numG1[], char numG2[], char resp[]);`

Esta función multiplica dos números grandes que se encuentran en una base dada y coloca la respuesta en otro arreglo. Si el número se desborda se devolverá 1 y en otro caso cero.

La operación se hará directamente en la base del número, no se permitirá convertir los números a otra base para realizar la operación. Debe respetar los signos de los números.

f) `int imprime(char numG1[]);`

Esta función imprime el número grande en el medio estándar. Sólo deben aparecer los dígitos significativos del número y el signo en caso de ser negativo.

PRIMERA PARTE (10 puntos): Creación de la biblioteca estática de funciones

Según lo indicado en las "instrucciones para el Laboratorio 2" que se publicó en la Intranet del curso, implemente empleando NetBeans las funciones mencionadas en un archivo denominado `numGBases.a` y sus correspondientes archivos `numGBases.h` y `numGBases.cpp`. Elabore un pequeño proyecto denominado `parte1` donde, de manera muy sencilla, se prueben todas funciones. Esta pregunta no se considerará como hecha si el JP no encuentra los archivos `.h` y `.cpp` en la carpeta.

SEGUNDA PARTE (10 puntos): ESTA PARTE SÓLO SE EVALUARÁ SI LA PRIMERA ESTÁ COMPLETA Y LA PRUEBA DA RESULTADOS CORRECTOS.

Haciendo uso obligatoriamente de la biblioteca `numGBases.a`, y del archivo `numGBases.h` (no podrá hacer uso aquí del archivo `numGBases.cpp`), elabore un proyecto denominado `parte2` que resuelva el siguiente problema:

Se tiene un archivo de textos, con la siguiente información:

```
8 35576324 + 562317 x -4056745213454 - -723452612
2 10001001010111 x 11111111111 + -1010101010101010
...
```

El archivo contiene, una expresión en cada línea. La línea empieza con el valor de la base en que están dados los números de la expresión. Los operadores que se podrán emplear en las expresiones son: + (suma), - (resta), x (multiplicación). No se considerará la precedencia de los operadores, la evaluación será de manera estricta de izquierda a derecha. Entre dato y dato podrá haber uno o más espacios en blanco o tabuladores.

El programa deberá mostrar el resultado de las expresiones en cada línea. Si alguno de números no corresponde con la base, se debe descartar. Si una operación provoca un desbordamiento se debe indicar en la respuesta. El reporte se debe presentar de una manera similar a la siguiente:

```
Resultado N° 1: -73612536526672 (8) No se descartaron valores.
Resultado N° 2: 100101010101111101100011 (2) Se descartaron 3 valores
Resultado N° 3: Se produjo un desbordamiento.
...
```

Anotaciones finales

Al finalizar el laboratorio, comprima¹ las dos carpetas creadas 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 `\Laboratorio2\<código del horario>\<aula>`.

La asignación de los puntajes estará fuertemente influenciada por el hecho que el programa presente errores de sintaxis y/o al ejecutar el programa no se obtenga resultados o estos estén errados.

No se calificará el código que esté dentro de comentarios.

Profesores: Miguel Guanira

San Miguel, 20 de septiembre del 2013.

¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip).