

# Metodología de la Programación

Convocatoria de Junio. Curso 2012/2013

24 de Junio de 2013

Se desea resolver el problema de sumar un número indeterminado de enteros no negativos, con la dificultad añadida de que dichos números pueden llegar a ser muy largos, siendo imposible usar el tipo *int* del lenguaje. Para resolverlo, se propone crear una clase *BigInt* (entero largo) que puede almacenar un entero no negativo de longitud indeterminada.

La clase representará un entero mediante un array -de longitud variable- de objetos de tipo *int*, reservado en memoria dinámica, para poder almacenar todos y cada uno de los dígitos de un entero de longitud indeterminada. Tenga en cuenta que un objeto *int* puede almacenar un rango muy amplio de valores, sin embargo, por simplicidad, será el tipo que usaremos para almacenar cada dígito -del 0 al 9- del *BigInt*.

Además, los dígitos del *BigInt* se almacenarán de forma que el menos significativo -las unidades- se situará en la posición cero del array. Por ejemplo, a continuación se presentan dos ejemplos: los enteros largos 9530273759835 y 0. Observe que para el primer caso se ha reservado y usado un array de objetos *int* de longitud 13 y, para el cero, un array de longitud uno, con el dígito 0.

0	1	2	3	4	5	6	7	8	9	10	11	12		0
5	3	8	9	5	7	3	7	2	0	3	5	9		0

Considerando este problema, **resuelva las siguientes preguntas:**

- (0.25 puntos)** Implemente el constructor sin parámetros y el destructor. Dicho constructor crea un entero largo con valor cero.
- Implemente dos constructores adicionales:
  - (0.75 puntos)** Constructor que crea un *BigInt* a partir de una cadena de caracteres. Esta cadena de caracteres contiene los dígitos del entero largo en formato decimal.
  - (1.5 puntos)** Constructor que crea un *BigInt* a partir de un objeto de tipo *unsigned int*.
- (1 punto)** Implemente el constructor de copia y operador de asignación.
- (2 puntos)** Sobrecargue el operador de suma para que, a partir de dos objetos *BigInt*, se obtenga un nuevo objeto que corresponde al entero largo resultado de su suma.
- (1 punto)** Sobrecargue el operador de salida (operador <<) para la clase *BigInt* de forma que nos permite escribir el entero largo en un flujo de salida. Tenga en cuenta que deberá presentar todos los dígitos desde el más significativo al menos significativo (escritura habitual de enteros).
- (1.5 puntos)** Sobrecargue el operador de entrada (operador >>) para la clase *BigInt* de forma que nos permite leer el entero largo desde un flujo de entrada. Tenga en cuenta que deberá leer todos los dígitos desde el más significativo al menos significativo. El operador de entrada debe comportarse de forma similar al caso del tipo *int*, es decir, asumiendo que cada *BigInt* -secuencia de dígitos consecutivos- se encuentra separado de otros datos por "espacios blancos" (espacios, tabuladores, saltos de línea, etc.).
- (1 punto)** Considerando todas las operaciones que se han descrito en los puntos anteriores, escriba el correspondiente archivo de cabecera ("bigint.h") teniendo en cuenta que no se incluye ninguna función en línea (*inline*). Es decir, sólo aparecerán cabeceras de funciones, sin su definición.
- (1 punto)** Considere que tiene correctamente implementadas las preguntas anteriores. Supongamos que tenemos un archivo con un número indeterminado de enteros largos en formato texto, separados por "espacios blancos". Implemente un programa "suma.cpp" que use la clase *BigInt* para leer todos los valores que hay en el fichero y escribir, como resultado final, la suma de todos ellos. Por ejemplo, si el archivo "datos.txt" contiene los siguientes enteros:

```
93478290374700000000000000000000
9327887198348931
```

Una posible ejecución del programa podría ser la siguiente:

```
% suma datos.txt
9347829037479327887198348931
```

**Duración del examen: 2 horas y media.**