

19 Dicembre 2016

Programmazione M-Z  
Ingegneria e Scienze Informatiche - Cesena  
A.A. 2016-2017

## Elaborato 10

**Data di sottomissione:** entro la mezzanotte dell'8 Gennaio 2017.

**Formato di sottomissione:** un file compresso con nome `Elaborato10.zip`, contenente un unico file sorgente con nome `bigint.c`

Specifiche:

- Sviluppare alcune funzioni di libreria basilari per poter eseguire operazioni matematiche su interi di lunghezza arbitraria.
- A meno di limitazioni legate alle risorse di memoria sulla macchina, la libreria deve permettere di manipolare interi di lunghezza arbitraria.
- La struttura dati utilizzata per rappresentare gli interi deve essere definita nell'implementazione. Unico vincolo: deve essere utilizzata una lista concatenata (in una variante qualsiasi).
- I prototipi delle funzioni da implementare sono dichiarati nell'header `bigint.h` e allegati alle specifiche.

Suggerimenti:

- Studiare attentamente la funzione di somma per poter individuare alcune caratteristiche della struttura `bigint` che possano semplificare lo sviluppo.
- Verificare i risultati dell'operazione di somma con i casi di test (`test.txt`) forniti con le specifiche.
- Prestare attenzione alla dichiarazione della struttura `BigInt` nel file header `bigint.h`. Tale struttura è dichiarata in modo *opaco*: i campi effettivi della struttura devono essere dichiarati nella libreria `bigint.c` e sono visibili unicamente alle funzioni presenti in tale file. Conseguenze: chi utilizza la libreria non potrà accedere a nessuno dei campi dell'oggetto `bigint`; l'unico modo per poter manipolare tali oggetti è tramite le funzioni di libreria.

```
1 #ifndef BIGINT_H
2 #define BIGINT_H
3
4 /*
5  * Forward declaration: the BigInt structure is implementation defined.
6  */
7 struct BigInt;
8
9 typedef struct BigInt bigint;
10
11 /*
12  * Converts the unsigned long int x into the bigint representation.
13  */
14 bigint *int2bigint(unsigned long int x);
15
16 /*
17  * Converts string representation of a positive integer number into
18  * the bigint representation.
19  *
20  * Returns a pointer to the bigint structure or NULL if the conversion
21  * cannot be performed.
22  */
23 bigint *str2bigint(char *str);
24
25 /*
26  * Converts a bigint number into a (printable) string.
27  *
28  * Returns a pointer to the first character of the string or NULL if
29  * the conversion cannot be performed.
30  */
31 char *bigint2str(bigint *num);
32
33 /*
34  * Returns the sum of the two input numbers or NULL if the operation
35  * cannot be performed.
36  */
37 bigint *sum(bigint *num1, bigint *num2);
38
39 /*
40  * Destroys the bigint data structure.
41  */
42 void clear(bigint **num);
43
44 #endif
```