BigInt

(Due Date: 01 July 2011)

One common limitation of built-in data types is that they typically have a limited number of values they can represent. Consider the int data type in C++, a typical implementation can only represent approximately 4 billion possible different numbers. In order to overcome this limitation, we will create a new data type called BigInt. Your task is to create a new class called BigInt that can represent arbitrarily large numbers (limited to only the amount of memory available to the program). To start with, your class should support 4 functions (add, subtract, multiply, divide). These member functions will take one arguments of type BigInt and perform their respective operations. In order to allow users to assign numbers to a BigInt object and output them, the class should also contain the member functions input and output. input() will take from standard input a string that represents a number. output() will output the number represented by a BigInt object to standard out.

For the BigInt class to hold a number, you may use a string object. For example:

```
string   _number = “12345678901234567890”;
```

To use string objects, make sure to #include <string>

To input a number from the standard input, you may use:

```
getline(cin, str_obj);
```

Some helpful information about strings:

any character can be accessed just like an array:

```
str_obj= “0123”;
cout << str_obj[3];  // prints 3

cout << str_obj.length()  // prints 4
```

To convert a character to int, you can use (int)(str_obj[2] – '0'); // prints 2

If you have any questions feel free to email me at stanovic@cs.fsu.edu