Create a project called daily7 and a source file called BigInt.cpp and a header file called BigInt.h.

Write a class called BigInt. The BigInt class is going to be able to hold and perform arithmetic on integers of arbitrary size (numbers with thousands, millions, or billions of digits). For this daily you are only doing a small portion of this class. Your class must contain a string internally to store the number since the BigInt can have millions of digits in it. You must also have a private member variable of type bool that remembers if the number is positive or negative.

```
class BigInt
{
public:
        BigInt(); // Initializes the BigInt to zero.
        BigInt(int x); //Initializes the BigInt to have the same value as x
        explicit BigInt(string x);//Initalizes the BigInt to have the value of the given string BUT
                                  //must check that the string is valid or exit(1) otherwise.
        friend ostream& operator<<(ostream& out, const BigInt& right);

private:
        string data;
        bool isNegative;
};
```

Write the three listed constructors. For the one that takes a string as an argument a valid string must , after an initial number of whitespace characters , begin with a digit, or + or -. Once the number begins it must only contain digits. If the string contains other information after the number is listed and a whitespace character is given then the string is still considered to be valid. For example, the following strings would be valid to the constructor:

“-100” – value negative 100

“+123123123123123123123123 dog” – value positive 123123123123123123123123

“ +123 + 123 + 123 +123 1231 123123123123” – value positive 123

Write the code for operator<< so that it works with the following main program:

```
int main()
{
        BigInt x;
        cout << x << endl;
        x = BigInt(42);
        cout << x << endl;
        x = BigInt("     -12345678901234567890123456789O and more stuff");
        cout << x << endl;
        x = BigInt("+-4848484848484848");
        cout << x << endl;
        return 0;
}
```

Turn in your BigInt.cpp and BigInt.h files on Blackboard. You do not need to turn in your main program as I will be using my own file to test your class. It is your responsibility to test it completely before turning it in.