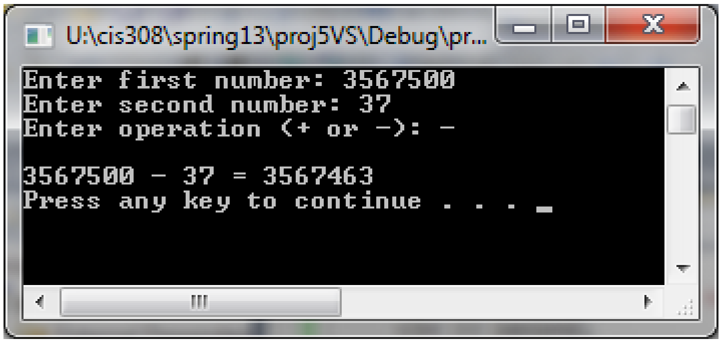
Programming Project 5 (30 points)
Due: Monday, April 1 by midnight

Assignment Description:

You are to write a program in C++ that performs arithmetic operations on integers (of possibly large size). You will prompt the user to input two numbers and an operation (+ or -). You will then print the corresponding expression and result. An example run of the program appears below.



```
Enter first number: 3567500
Enter second number: 37
Enter operation (+ or -): -

3567500 - 37 = 3567463
Press any key to continue . . . _
```

Implementation Requirements:

Your program must include the following three files:

- `biginteger.h`
- `biginteger.cpp`
- `proj5.cpp`

`biginteger.h` should contain the definition for `class BigInteger`. This class represents an integer of indeterminate size (e.g., 3567500 or 7928375134987460). Because an int (and other numerical types) can only represent numbers of a limited size, you will use a string instance variable to store the digits of the number. It should have a constructor to initialize the number as well as the following public function definitions:

- `BigInteger plus(BigInteger)` – return a new `BigInteger` object that represents THIS number plus the `BigInteger` argument

- `BigInteger minus(BigInteger)` – return a new `BigInteger` object that represents THIS number minus the `BigInteger` argument

- `string toString(void)` – returns the string representation of this number

`biginteger.cpp` should implement each function in `biginteger.h`. You are welcome to include more functions if you want.

`proj5.cpp` should contain the `main` function. It should get the user input for the two numbers and operator, create two `BigInteger` objects, apply the operator by calling a function in `BigInteger`, and print the expression and result to the screen. It should call the `toString` function in `BigInteger` to print the value of a number. Your prompts and output should exactly match the example above, and you should print an error and exit if the user enters an invalid operator.

You may assume that the user will only enter positive numbers, and that the result of the expression (even after subtraction) will still be positive.

You should write your program in Visual Studio. When you submit your assignment, find where you stored your project on your computer. Zip the entire project folder (which should include the Visual Studio Solution file and all your source code) and submit that online.

Extra Credit (5 pts):

You may earn 5 points of extra credit by extending your solution to work with negative inputs and negative results. For example, -376 – 7112 should yield -7488.

Documentation:

Your program must include a comment block at the top of every file and every function. The function comments should include a brief description of what the function does, and explain any function arguments and return values. You may use the comment block below as a template:

```
/*********************************************
* Name: (YOUR NAME)                          *
* Date: (THE DUE DATE)                       *
* Assignment: Project 5: Big Integers        *
**********************************************
* (WRITE A DESCRIPTION OF THE PROGRAM)       *
**********************************************/
```

Submission:

Your project must be submitted using the Project Submission Link on K-State Online. Submit a zip file of your project using this submission page.

Grading:

Programs that do not build in Visual Studio will receive a grade of 0. A grading breakdown for programs that do compile appears below:

`BigInteger` class (`biginteger.h`)	4
`plus`, `minus` (separate functions, correct result)	12
`toString` function	4
Input and output (correct format)	2
Create `BigInteger` objects	2
Correct program structure (`biginteger.h`, `biginteger.cpp`, `proj5.cpp`)	2
Visual Studio Solution	2
Documentation	2
Extra credit (works with negative inputs and negative results)	5
Total	**30**