3460:4/521 Project 2 Report

Name: Rob Bauer
UANET id: rsbauer

Generated: Mon Nov 8 16:00:11 EST 2010

```
make clean
rm -f BigInt_s BigInt_s.o BigInt_t BigInt_t.o BigInt.o vadd.o

make
g++ -c BigInt_s.cpp
g++ -c vadd.cpp
g++ -c BigInt.cpp
g++ BigInt_s.o vadd.o BigInt.o -o BigInt_s

make test
g++ -c BigInt_t.cpp
g++ BigInt_t.o BigInt.o vadd.o -o BigInt_t
./BigInt_t

make clean
rm -f BigInt_s BigInt_s.o BigInt_t BigInt_t.o BigInt.o vadd.o
```

rsbauer/Projects/Project2/BigInt/BigInt.hpp:

```
/*
        BigInt.hpp

        BigInt functions.  Perform infinite precision arithmetic.

        Rob Bauer
        rbauer@tekro.com
*/

#ifndef INCLUDED_BIGINT_HPP
#define INCLUDED_BIGINT_HPP

#include "vadd.hpp"
#include <iostream>
#include <vector>

class BigInt
{
private:
        std::vector<int> digits;

public:
        // Default constructor
        BigInt();

        // Copy constructor
        BigInt(const BigInt &);

        // Initialize BigInt with an specified integer
        BigInt(const int);

        // Initialize BigInt with a string
        BigInt(const std::string &);

        // Initialize BigInt with char *
        BigInt(const char *);

        // get the number of digits stored
        int size();

        // convert a string to digits
        void convertString(const char *);

        // Stream output
        friend std::ostream& operator<<(std::ostream &, const BigInt &);

        // add numbers using +=
        const BigInt& operator+=(const BigInt &);

        // add numbers using +
        friend BigInt operator+(const BigInt &, const BigInt &);

        // test BigInt to a string
        bool operator==(const std::string &);

        // convert BigInt to string
        std::string toString();

};

// Stream output
std::ostream& operator<<(std::ostream &, const BigInt &);

// adding
BigInt operator+(const BigInt &, const BigInt &);

#endif
```

rsbauer/Projects/Project2/BigInt/BigInt.cpp:

```
/*
        BigInt.cpp

        BigInt function definitions.  Perform infinite precision arithmetic.

        Rob Bauer
        rbauer@tekro.com
*/

#include "BigInt.hpp"
#include "vadd.hpp"
#include <iostream>
#include <sstream>
#include <vector>


// Default constructor
BigInt::BigInt()
{
}


// Copy constructor
BigInt::BigInt(const BigInt &bint)
{
        digits = bint.digits;
}

// Initialize BigInt with an integer
BigInt::BigInt(const int number)
{
        int value = number;
        while(value > 0)
        {
                digits.push_back(value % 10);
                value = value / 10;
        }
}

// Initialize BigInt with a string
BigInt::BigInt(const std::string &str)
{
        convertString(str.c_str());
}

// Initialize BigInt with a char *
BigInt::BigInt(const char *str)
{
        convertString(str);
}

// return the size of digits
int BigInt::size()
{
        return digits.size();
}

// convert a string to digits
void BigInt::convertString(const char *str)
{
        int num = 0;

        while(*str != '\0')
        {
                // get one character at a time
                num = *str++;

                // make sure the char is in numeric range
                if(num >= '0' && num <= '9')
                {
                        // convert to int
                        num -= 48;

                        // add to the digits list
                        digits.insert(digits.begin(), 1, num);
                }
        }
}


// Stream output
std::ostream& operator<<(std::ostream &out, const BigInt &bint)
{
        for(int a = bint.digits.size() - 1; a > -1; a--)
        {
                out << bint.digits[a];
        }

        return out;
}
```

3460:4/521 Project 2 Report

```
// add numbers using +=
const BigInt& BigInt::operator+=(const BigInt &right)
{
        add(this->digits, right.digits);
        return *this;
}


// add numbers using + syntax
BigInt operator+(const BigInt &left, const BigInt &right)
{
        BigInt target(left);
        target += right;


        return target;
}

// test BigInt against a string
bool BigInt::operator==(const std::string &right)
{
        return this->toString() == right;
}

// convert a BigInt to a string
std::string BigInt::toString()
{
        std::ostringstream sout;
        sout << *this;
        return sout.str();
}
```

rsbauer/Projects/Project2/BigInt/BigInt_t.cpp:

```
/*
  BigInt_t.cpp

  BigInt unit tests

  Rob Bauer
  rbauer@tekro.com
*/

#include "BigInt.hpp"
#include <cassert>
#include <iostream>
#include <sstream>

void testOutput();
void testAdding();
void testComparison();
void testStringStream(const BigInt &, const char *);

int main()
{
        BigInt b;
        BigInt b2(b);
        BigInt b3(123);
        BigInt b4("123");

        // a silly test to verify the addresses are not the same
        assert(&b2 != &b);


        b = b2;
        assert(&b != &b2);
        assert(b == "");

        // not able to test much other than verify the proper constructor is firing
        b = "123";

        // test the size
        assert(b3.size() == 3);         // init with int
        assert(b3 == "123");

        assert(b.size() == 3);          // init with str
        assert(b == "123");

        // test copy constructor
        BigInt b5(b4);
        assert(b5.size() == 3);
        assert(b5 == "123");

        BigInt b6(b3);
        assert(b6.size() == 3);
        assert(b6 == "123");

        // test really big numbers
        BigInt big("12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890");
        assert(big.size() == 250);

        // test bad input
        BigInt bad("asdf");
        assert(bad.size() == 0);

        testOutput();

        testAdding();

        testComparison();

        return 0;
}


// Test output, both cout << and to a file
void testOutput()
{
        BigInt b;

        // test empty BigInt
        testStringStream(b, "");

        // test string
        b = "123";
        testStringStream(b, "123");

        // test int
        b = 456;
        testStringStream(b, "456");

}

// test string stream
void testStringStream(const BigInt &bint, const char *expected)
{
        std::ostringstream sout;
        sout << bint;
        assert(sout.str() == expected);
}


// test adding BigInts together
void testAdding()
{
        BigInt add1(9999);
        BigInt add2(1);
        BigInt add3 = add1 + add2;

        assert(add3.size() == 5);
        assert(add3.toString() == "10000");
        assert(add3 == "10000");

        add1 += add2;
        assert(add1.size() == 5);
        assert(add1.toString() == "10000");
        assert(add1 == "10000");
        assert(add2.size() == 1);
        assert(add2 == "1");
}

// test comparison operators
// right now, only == supported
void testComparison()
{
        BigInt a(123);

        assert(a == "123");
        assert(a.toString() == "123");
}
```

rsbauer/Projects/Project2/BigInt/Makefile:

```
##
# BigInt Make

## Executable ##
all : BigInt_s

BigInt_s : BigInt_s.o vadd.o BigInt.o
        g++ BigInt_s.o vadd.o BigInt.o -o BigInt_s

BigInt_s.o : BigInt.cpp BigInt.hpp BigInt_s.cpp
        g++ -c BigInt_s.cpp

vadd.o : vadd.cpp vadd.hpp
        g++ -c vadd.cpp

BigInt.o : BigInt.cpp BigInt.hpp
```

3460:4/521 Project 2 Report

```
	g++ -c BigInt.cpp

## Tests ##
BigInt_t.o : BigInt_t.cpp BigInt.hpp BigInt.cpp
	g++ -c BigInt_t.cpp

BigInt_t : BigInt_t.o BigInt.o
	g++ BigInt_t.o BigInt.o vadd.o -o BigInt_t

test : BigInt_t
	./BigInt_t


## House Cleaning ##
clean :
	rm -f BigInt_s BigInt_s.o BigInt_t BigInt_t.o BigInt.o vadd.o

------------------------------------------------------------------------
r1148 | rsbauer | 2010-10-25 17:07:44 -0400 (Mon, 25 Oct 2010) | 1 line

Added additional unit tests to test bad input and really big numbers
------------------------------------------------------------------------
r1077 | rsbauer | 2010-10-24 16:44:56 -0400 (Sun, 24 Oct 2010) | 1 line

Refactored and added +=, ==, and + operators. Added and updated tests.
------------------------------------------------------------------------
r874 | rsbauer | 2010-10-15 06:57:44 -0400 (Fri, 15 Oct 2010) | 1 line

Refactored and added tests. Found bugs and fixed. Tests passing.
------------------------------------------------------------------------
r805 | rsbauer | 2010-10-10 15:38:12 -0400 (Sun, 10 Oct 2010) | 1 line

Added size method for r728. Updated Makefile to fix bug. Added code to constructors to support size()
------------------------------------------------------------------------
r804 | rsbauer | 2010-10-10 14:24:21 -0400 (Sun, 10 Oct 2010) | 1 line

Added BigInt constructor for handling char * in regards to r722
------------------------------------------------------------------------
r803 | rsbauer | 2010-10-10 12:47:02 -0400 (Sun, 10 Oct 2010) | 1 line

Added constructor to handle strings for r721 and updated unit tests
------------------------------------------------------------------------
r802 | rsbauer | 2010-10-10 12:44:41 -0400 (Sun, 10 Oct 2010) | 1 line

Added BigInt(int) constructor, default, and copy constructor for r720. Updated unit tests. Added comments.
------------------------------------------------------------------------
r798 | rsbauer | 2010-10-09 16:37:19 -0400 (Sat, 09 Oct 2010) | 1 line

Verified operator overload works for r719.  Cleaned up white space
------------------------------------------------------------------------
r797 | rsbauer | 2010-10-09 16:35:04 -0400 (Sat, 09 Oct 2010) | 1 line

Verified << overload works for r718 and added unit tests. Makefile wasn't including BigInt.o for BigInt_t and has be
en updated.
------------------------------------------------------------------------
r796 | rsbauer | 2010-10-09 16:22:28 -0400 (Sat, 09 Oct 2010) | 1 line

Added cout << by overloading << operator for r717. Updated Makefile - was missing BigInt.o references.
------------------------------------------------------------------------
r795 | rsbauer | 2010-10-09 15:53:50 -0400 (Sat, 09 Oct 2010) | 1 line

Added test for r716 and fixed previous merge issue (previous check had blank due to trying to fix a merge
------------------------------------------------------------------------
r700 | rsbauer | 2010-10-02 21:47:03 -0400 (Sat, 02 Oct 2010) | 1 line


------------------------------------------------------------------------
r698 | rsbauer | 2010-10-01 21:58:46 -0400 (Fri, 01 Oct 2010) | 1 line

Added vadd.cpp/hpp to the build (using the switched in version)
------------------------------------------------------------------------
r697 | rsbauer | 2010-10-01 17:28:52 -0400 (Fri, 01 Oct 2010) | 1 line

Setup the BigInt class enough to be able to compile BigInt_s r684 and r687. Setup Makefile, BigInt.cpp. Added a simp
le unit test to BigInt_t
------------------------------------------------------------------------
r688 | collard | 2010-09-30 20:03:36 -0400 (Thu, 30 Sep 2010) | 1 line

Setup for vadd base functions
------------------------------------------------------------------------
r685 | collard | 2010-09-30 18:26:53 -0400 (Thu, 30 Sep 2010) | 1 line

Setup for Project 2
------------------------------------------------------------------------
```