```
/*
 * These functions are designed to help you test your MyString objects,
 * as well as show the client usage of the class.
 *
 * The BasicTest function builds an array of strings using various
 * constructor options and prints them out.  It also uses the String
 * stream operations to read some strings from a data file.
 *
 * The RelationTest function checks out the basic relational operations
 * (==, !=, <, etc) on Strings and char *s.
 *
 *
 * The CopyTest tries out the copy constructor and assignment operators
 * to make sure they do a true deep copy.
 *
 * Although not exhaustive, these tests will help you to exercise the basic
 * functionality of the class and show you how a client might use it.
 *
 * While you are developing your MyString class, you might find it
 * easier to comment out functions you are ready for, so that you don't
 * get lots of compile/link complaints.
*/

#include "mystring.h"
#include <cctype>      // for toupper()
#include <iostream>
#include <string>
using namespace std;
using namespace cs_mystring;

void BasicTest();
void RelationTest();
void CopyTest();
MyString AppendTest(const MyString& ref, MyString val);
string boolString(bool convertMe);

int main()
{
    BasicTest();
    RelationTest();
    CopyTest();
}




void BasicTest()
{
    MyString s;
    cout << "----- Testing basic String creation & printing" << endl;

    const MyString strs[] =
                {MyString("Wow"), MyString("C++ is neat!"),
                 MyString(""), MyString("a-z")};

    for (int i = 0; i < 4; i++){
        cout << "string [" << i <<"] = " << strs[i] << endl;
    }

    cout << endl << "----- Testing access to characters (using const)" << endl;
    const MyString s1("abcdefghijklmnopqsrtuvwxyz");
    cout <<  "Whole string is " << s1 << endl;
    cout << "now char by char: ";
```

```
    for (int i = 0; i < s1.length(); i++){
        cout << s1[i];
    }

    cout << endl << "----- Testing access to characters (using non-const)" << endl;
    MyString s2("abcdefghijklmnopqsrtuvwxyz");
    cout <<  "Start with " << s2;
    for (int i = 0; i < s2.length(); i++){
        s2[i] = toupper(s2[i]);
    }
    cout << " and convert to " << s2 << endl;
}




string boolString(bool convertMe) {
    if (convertMe) {
        return "true";
    } else {
        return "false";
    }
}




void RelationTest()
{
    cout << "\n----- Testing relational operators between MyStrings\n";

    const MyString strs[] =
        {MyString("app"), MyString("apple"), MyString(""),
        MyString("Banana"), MyString("Banana")};

    for (int i = 0; i < 4; i++) {
        cout << "Comparing " << strs[i] << " to " << strs[i+1] << endl;
        cout << "    Is left < right? " << boolString(strs[i] < strs[i+1]) << endl;
        cout << "    Is left <= right? " << boolString(strs[i] <= strs[i+1]) << endl;
        cout << "    Is left > right? " << boolString(strs[i] > strs[i+1]) << endl;
        cout << "    Is left >= right? " << boolString(strs[i] >= strs[i+1]) << endl;
        cout << "    Does left == right? " << boolString(strs[i] == strs[i+1]) << endl;
        cout << "    Does left != right ? " << boolString(strs[i] != strs[i+1]) << endl;
    }

    cout << "\n----- Testing relations between MyStrings and char *\n";
    MyString s("he");
    const char *t = "hello";
    cout << "Comparing " << s << " to " << t << endl;
    cout << "    Is left < right? " << boolString(s < t) << endl;
    cout << "    Is left <= right? " << boolString(s <= t) << endl;
    cout << "    Is left > right? " << boolString(s > t) << endl;
    cout << "    Is left >= right? " << boolString(s >= t) << endl;
    cout << "    Does left == right? " << boolString(s == t) << endl;
    cout << "    Does left != right ? " << boolString(s != t) << endl;

    MyString u("wackity");
    const char *v = "why";
    cout << "Comparing " << v << " to " << u << endl;
    cout << "    Is left < right? " << boolString(v < u) << endl;
    cout << "    Is left <= right? " << boolString(v <= u) << endl;
    cout << "    Is left > right? " << boolString(v > u) << endl;
    cout << "    Is left >= right? " << boolString(v >= u) << endl;
```

```
    cout << "    Does left == right? " << boolString(v == u) << endl;
    cout << "    Does left != right ? " << boolString(v != u) << endl;
}




MyString AppendTest(const MyString& ref, MyString val)
{
    val[0] = 'B';
    return val;
}




void CopyTest()
{
    cout << "\n----- Testing copy constructor and operator= on MyStrings\n";

    MyString orig("cake");

    MyString copy(orig);    // invoke copy constructor

    copy[0] = 'f';  // change first letter of the *copy*
    cout << "original is " << orig << ", copy is " << copy << endl;

    MyString copy2;      // makes an empty string

    copy2 = orig;        // invoke operator=
    copy2[0] = 'f';      // change first letter of the *copy*
    cout << "original is " << orig << ", copy is " << copy2 << endl;

    copy2 = "Copy Cat";
    copy2 = copy2;        // copy onto self and see what happens
    cout << "after self assignment, copy is " << copy2 << endl;

    cout << "Testing pass & return MyStrings by value and ref" << endl;
    MyString val = "winky";
    MyString sum = AppendTest("Boo", val);
    cout << "after calling Append, sum is " << sum << endl;
    cout << "val is " << val << endl;
    val = sum;
    cout << "after assign,  val is " << val << endl;
}
```