Overview

- Test 1 will be held **Tuesday, February 8, 2005, 10:00-11:50am, Sage 3303.** No make-ups will be given except for emergency situations, and even then a written excuse from Dean of Students office will be required.

- **Purpose:** Check your understanding of the basics of (a) solving small computational problems, (b) C++, and (c) the standard library.

- **Coverage:** Lectures 1-5, Labs 1-3, HW 1-2.

- Closed-book and closed-notes. Photocopies of the “Details” sections from Chapters 0-4, 9 of the text will be provided.

- Below are relevant sample questions from previous tests. Solutions will be posted on-line.

- **How to study?**
  - Review lecture notes
  - Review and re-do lecture exercises, lab and homework problems.
  - Do the practice problems. Practice writing solutions using pencil (or pen) and paper.

- At least one of these questions will appear on the test.

- On the other hand, this set of questions is not necessarily comprehensive because our emphasis has been slightly different this semester than previous semesters.

Practice Problems

1. Write a code segment that copies the contents of a string into a vector of char in reverse order.

2. Write a function that takes a vector of strings as an argument and returns the number of vowels that appear in the string. A vowel is defined as an ’a’, ’e’, ’i’, ’o’ or ’u’. For example, if the vector contains the strings
lincoln went to the white house

Your function should return the value 12.
You may assume that all letters are lower case. Here is the function prototype:

```cpp
int count_vowels( const vector<string>& strings )
```

3. Write a function called `less_string` that mimics the effect of the `<` operator on strings. In other words, given strings `a` and `b`, `less_string(a, b)` should return `true` if and only if `a < b`. Of course, you may use `<` on individual characters in the string. Start by getting the function prototype correct.

Here are examples of pairs for which your function should return `true`:

```cpp
a = "abc",  b = "abd"
a = "cab",  b = "cabbage"
a = "christine",  b = "christopher"
```

4. What is the output from the following program? We strongly suggest that you draw the contents of the vectors to help you visualize what is happening.

```cpp
void more_confused( vector<int> a, vector<int> & b )
{
    for ( unsigned int i=0; i<2; ++i )
    {
        int temp = a[i];
        a[i] = b[i];
        b[i] = temp;
    }
    cout << "i: ";
    for ( unsigned int i=0; i<a.size(); ++i )
```
cout << a[i] << " ";
cout << endl;
cout << "2: ";
for ( unsigned int i=0; i<b.size(); ++i )
    cout << b[i] << " ";
cout << endl;
}

int
main()
{
    vector<int> a, b;
a.push_back(1); a.push_back(3); a.push_back(5);
b.push_back(2); b.push_back(4);

    more_confused( a, b );
a[0] = 7; a[1] = 9;
    more_confused( b, a );

    cout << "3: ";
    for ( unsigned int i=0; i<a.size(); ++i )
        cout << a[i] << " ";
cout << endl;

    cout << "4: ";
    for ( unsigned int i=0; i<b.size(); ++i )
        cout << b[i] << " ";
cout << endl;

    return 0;
}

5. Write a function that takes a vector of doubles and copies its values into two vectors of doubles, one containing only the negative numbers from the original vector, the other containing only the positive numbers. Values that are 0 should not be in either vector. For example, if the original vector contains the values

-1.3, 5.2, 8.7, -4.5, 0.0, 7.8, -9.1, 3.5, 6.6

then the resulting vector of negative numbers should contain

-1.3, -4.5, -9.1
and the resulting vector of positive numbers should contain

5.2, 8.7, 7.8, 3.5, 6.6

Start this problem by writing the function prototype as you think it should appear and then write the code.