Assignment: Data Structure

Sparse matrix is a matrix of which most of the elements are zeros. To save the storage, we can use several format to store sparse matrices. Those formats include Compressed Row Storage, Compressed Column Storage, etc. See here: http://netlib.org/linalg/html_templates/node90.html.

Now we are going to use a simpler format. In this format, we only store the non-zero elements. Each element is characterized by <row, col, value>.

The matrix is characterized by a class SparceMatrix, with the usage of vector in STL.

The declaration code should be:

```
#include <vector>
using namespace std;

struct term {
        int col;
        int row;
        double value;
};

class SParseMatrix:public vector<term>
{
        int ncol;
        int nrow;
};
```

Here we set SparseMatrix as a derived class of vector<term>. And thus the SparseMatrix inherits lot of functions directly from vector<term> class, including the constructor, copy construct, assignment operator.

Following that, your work is listed as following:

1. Create either member functions or constructors to initialize "ncol" and "nrow", the number of column and row.
2. Complete the SparseMatrix by adding the operator+ function, add two sparse matrix together.
3. Add the operator* function, SparseMatrix times SparseMatrix, return another SparseMatrix.
4. Create main function to test the two operators.

Zip your code, output and any documents and submit into the Blackboard.