

Jan 08, 04 11:15	sortdemo.cpp	Page 1/2
------------------	---------------------	----------

```

#include <iostream>
#include <fstream>
#include <map>
#include <vector>
#include <algorithm>    // for exit
#include <set>
#include <string>
#include <sstream>
#include <cstdlib>      // for exit
using namespace std;

#include "ctimer.h"

/**
 * Show some STL sort functions and overloaded operators
 *
 * @author Owen Astrachan
 */

ostream& operator <<(ostream& out, const pair<string,int>& p)
// post: string and int printed, int first
{
    out << p.second << "\t" << p.first;
    return out;
}

class WordTracker
{
public:
    WordTracker();
    virtual void read(const string& filename);
    virtual void top(int n, vector<pair<string,int> >& v);

protected:
    map<string,int> myMap;
};

WordTracker::WordTracker()
{
    // nothing to do currently
}

void WordTracker::read(const string& filename)
{
    ifstream input(filename.c_str());
    string word;
    while (input >> word){
        myMap[word]++;
    }
}

bool sortoccurs(const pair<string,int>& lhs,
               const pair<string,int>& rhs)
{
    return lhs.second > rhs.second;
}

void WordTracker::top(int n, vector<pair<string,int> >& v)
{
    v.clear();

    // back_insert_iterator<vector<pair<string, int > > > bvi(v);
    // copy(myMap.begin(), myMap.end(),bvi);

    map<string,int>::iterator it = myMap.begin();

```

Jan 08, 04 11:15	sortdemo.cpp	Page 2/2
------------------	---------------------	----------

```

    while (it != myMap.end()){
        v.push_back(*it);
        it++;
    }

    partial_sort(v.begin(), v.begin()+n, v.end(), sortoccurs);
    // sort(v.begin(), v.end(),sortoccurs);
}

int main(int argc, char *argv[])
{
    string filename,line, w;
    if (argc == 1)
    {
        cerr << "usage: " << argv[0] << " filename" << endl;
        exit(1);
    }
    filename = argv[1];
    WordTracker tracker;

    CTimer timer;
    timer.Start();
    tracker.read(filename);
    vector<pair<string,int> > v;
    tracker.top(50,v);
    timer.Stop();

    for(int k=0; k < 53; k++){
        cout << v[k] << endl;
    }
    cout << "time = " << timer.ElapsedTime() << endl;
    return 0;
}

```