2、 有一文本文件(scores.txt)包含的内容如下

```
#name        subject 1 … subject 8
appropriated 89  83  94  79  89  92  94  94
enfeebling   89  83  89  92  75  91  92  89
instancing   88  84  92  87  71  90  94  93
meticulous   89  83  94  94  91  80  94  94
…
```

编写程序，完成如下功能：读取文本文件的内容，并计算平均分，按照总分从大到小的顺序排列，并把结果输出到文件(scoresort.txt)中。输出文件内容如下：

```
#name        avg    subject 1 … subject 8
meticulous   87.6   89  83  94  94  91  80  94  94
appropriated 87.4   89  83  94  79  89  92  94  94
instancing   87     88  84  92  87  71  90  94  93
enfeebling   86.8   89  83  89  92  75  91  92  89
```

参考代码：

```
#include <iostream>
#include <vector>
#include <string>
#include <fstream>

class Student{
public:
    std::string name;
    std::vector<int> scores;
    int total;
    float avg;
    Student(std::vector<std::string> words){
        std::vector<std::string>::iterator it = words.begin();
        name = *it;
        ++it;
        total = 0;
        while(it!=words.end()){
            int sc = atoi( (*it).c_str() );
            scores.push_back(sc);
            ++it;
            total += sc;
        }
        avg = total /scores.size();
    }

    friend std::ostream& operator<<(std::ostream& os, const Student& astu);
```

```cpp
};

std::ostream& operator<<(std::ostream& os, const Student& astu)
{
    std::string SEP="\t";
    os<< astu.name << SEP << astu.avg << SEP;
    for(int i=0;i<astu.scores.size();i++)
        os<<astu.scores[i] << SEP;
    return os;
}

class Students{
    std::vector<Student> items;
public:
    Students(){
        std::string line;
        std::ifstream fin("z:\\scores.txt");

        while(getline(fin,line)){
            std::vector<std::string> words = Split(line,"\t");
            items.push_back(Student(words));
        }
        fin.close();
    }

    void Sort(){
        if (items.size()<2) return;

        for(size_t i=0;i<items.size();++i){
            for(size_t j=i+1;j<items.size();++j)
            {
                if(items[i].total<items[j].total){
                    Student temp = items[i];
                    items[i] = items[j];
                    items[j] = temp;
                }
            }
        };
    };

    std::vector<std::string> Split(const std::string& s, const std::string& c){
        std::vector<std::string>  v;
        std::string::size_type pos1, pos2;
        pos2 = s.find(c);
        pos1 = 0;
        while(std::string::npos != pos2){
            v.push_back(s.substr(pos1, pos2-pos1));
            pos1 = pos2 + c.size();
            pos2 = s.find(c, pos1);
        }
        if(pos1 != s.length())
            v.push_back(s.substr(pos1));
        return v;
    }

    friend std::ostream& operator<<(std::ostream& os, const Students& list);
};

std::ostream& operator<<(std::ostream& os, const Students& list)
{
    for(size_t i=0;i<list.items.size();++i)
        os<<list.items[i] << "\n";
    return os;
}
```

考核要点：

1)书写多个完整类的能力；

2)文件操作；

3)输出重载；

4)基本算法：排序；

5)基本输入输出组织能力。