CSE 241 Assignment #4 - Assignment Scores Spring 2014

Due date: April 4 (Fri) - 20:00

AssignmentScores Class

Implement a class that computes the total sum, maximum, and minimum of assignment scores for CSE241 class, and also for an individual student. Your program must implement the below"AssignmentScores" class and a main function for this problem. You MUST NOT change the given class declaration, including the names of member function parameters and member variables. You MUST NOT add any other member variables and functions.

- The max number of students registered for CSE241 is 100.
- The max number of assignments for CSE241 is 10.
- In *setScores* function, the scores of a student are stored in *scores* variable, which is a dynamically allocated array. For each assignment, if a student submits the assignment, a score s, where $0 \leq s \leq 100$, is graded and stored in *scores* variable. If a student does not submit the assignment, the corresponding score, that is 0, will not be stored in *scores* variable. Thus, the size of the array is the same as the number of assignments submitted by a student.
- The maximum assignment score is the largest score graded for any assignment. Similarly, the minimum assignment score is the smallest score graded for any assignment.

```
class AssignmentScores
{
private:
   static int totalSum;    // total sum of assignment scores for class
   static int overallMax;  // max assignment score for class
   static int overallMin;  // min assignment score for class

   string name;            // name of a student
   int num;                // number of assignments submitted by this student
   int sum;                // sum of assignment scores of this student
   int* scores;            // array of assignment scores of this student

public:
   static int getTotalSum();          // get total sum of scores for class
   static int getOverallMin();        // get overall max score
   static int getOverallMax();        // get overall min score

   AssignmentScores operator=(const AssignmentScores &right); // overloaded operator
   AssignmentScores();                // constructor
   AssignmentScores(const AssignmentScores &assignScore); // copy constructor
   ~AssignmentScores();               // destructor
   void setName(string name);         // set name of this student
   string getName();                  // get name of this student
   void setNum(int num);              // set num of assignments submitted by this student
   int getSum();                      // get sum of scores for this student
   int getMax();                      // get max scores of this student
   int getMin();                      // get min scores of this student
   void setScores(int scores[], int num);  // set assignment scores of this student
};
```

Input

The input begins with a single positive integer k (where $0 < k \leq 100$) on a line by itself indicating the number of registered students in CSE241. The next k lines of input contain the assignment scores for k students. Each of the k lines consists of the name of a student, the number of assignments submitted by this student, n (where $0 \leq n \leq 10$), and a list of n scores if $n > 0$.

Output

The output will consist of the total sum of assignment scores for all students in CSE241, the max assignment score for CSE241, and the min assignment score for CSE241 on each line. After then, for each of k students, print a line giving the name of a student, the sum of assignment scores for the student, the max assignment score of the student, and the min assignment score of the student.

Sample Input

```
3
Alice 8 80 90 90 90 70 90 90 70
Bob 10 100 90 90 90 80 100 80 90 80 80
David 5 50 70 90 80 80
```

Sample Output

```
1920
100
50
Alice 670 90 70
Bob 880 100 80
David 370 90 50
```

Submitting Your Code and Report

Your file name must be "assign4.cpp", and it is the only file you should submit. Please do not submit the executable file of your program. Turn in your project using the "oopsubmit" command as follows:

```
$ oopsubmit assign4 assign4.cpp
```

Late submissions will lose 20 points per day.

If you do not follow this submission guideline, you will lose 10 points out of 100.

You should also submit a hard copy of your code to TA. Your report must have a cover page with your student ID and name. In the report, your code must be well commented to explain your algorithm. Also, the sample input and the output of your program must be included in the report.