COP 3223H Honors Intro to Programming - Program #7

Due date: Please consult WebCourses for your section

Objectives

1. To practice writing a program in C++.
2. To use arrays in a program.
3. To design a functional breakdown to solve a problem.

Problem: UCF Idol Extension (ucfidol2.cpp)

The UCF Idol Show has been a hit. But, some of the audience members have been complaining because some of the judges (in the audience), have been biasing their scores to help out their friends who are competing.

In particular, several cast members from Universally Simple Farmland all go to the same UCF Idol Show (because, frankly, it's better than theirs) and when they judge, they give very high scores to their cronies and very low scores to everyone else.

After enough of their shenanigans, management had gotten enough complaints that they want you to do something about these hooligans. However, the public relations department has advised against kicking out any paying patrons.

Your boss has finally come up with an idea whereby these judges can't destroy the fairness of the UCF Idol competition. In particular, she had noticed that the "cheating" judges tend to give scores that are spread out a great deal, while regular judges do not. She vaguely remembers from her statistics class that standard deviation is a rough measure of how far spread out a set of data is. Her idea is to simply calculate the standard deviation of each judge's scores, and then throw out ALL of a judge's scores if this standard deviation exceeds some chosen constant. (The operations people will track the show and determine this value on a day to day basis.)

Of course, the danger in this technique is that all the scores may be thrown out. In this special case, nothing can be done and all the scores will be used.

Your job will be to write a program that processes the scores for the UCF Idol show in this manner, adjusting the final averages for contestants based on the "fairness" of the judges.

Note: Given a set of numbers a_1, a_2, a_3, … , a_n, let the average of these numbers be A. (Namely, $A = \frac{a_1+a_2+a_3+\cdots+a_n}{n}$.) The standard deviation for this set of numbers is defined as follows:

$$Std\ Dev(a_1, a_2, a_3, \ldots, a_n) = \sqrt{\frac{(A-a_1)^2 + (A-a_2)^2 + (A-a_3)^2 + \cdots + (A-a_n)^2}{n}}$$

Input Format (read from stdin, no print prompts!!!)

The first line of input will contain a single positive integer, *n(1 < n < 100)*, representing the number of shows that are being scored. The data for each show follows, starting on the next line.

The first line of each show will contain three integers separated by spaces: *C (1 < C < 10)*, representing the number of contestants, *A (2 < A < 101)*, representing the number of audience members judging the contestants, and *S (5 < S < 100)*, representing the threshold standard deviation for that case. All judges who have a standard deviation above *S* should have their scores thrown out, unless this is true of all judges. In that case, all the scores stay.

The next *C* lines will contain *A* positive integers separated by spaces in between 1 and 100, inclusive, representing all of the scores of that particular contestant from each of the judges, in order. (The first line has the scores for contestant number 1, the second line for contestant number 2, etc.)

Output Specification (to the screen)

For each show, output a header with the following format on a line by itself:

```
UCF IDOL SHOW #k
```

where k will be the number of the show, starting at 1.

The next line of output for each show will output the winning contestant number as well as their average score:

```
Contestant #m, you win with an average score of X.
```

Note: m will be a positive integer in between 1 and the number of volunteers. X will be a real number in between 1 and 100 printed to 2 decimal places.

If none of the judge scores were thrown out, the last line of output for the case should read (this happens when all scores are above S or all scores are below S):

```
All judges' scores were used.
```

If at least one judge's scores were thrown out, output a line with the following format:

```
The following judges' scores were thrown out: A B … D
```

where A, B, etc. represent the number of each judge (starting at 1), who had their scores thrown out. Each of these judge numbers must be outputted in increasing order.

Place 2 blank lines between the output for shows.

Note: You are guaranteed that only one contestant will have the highest average.

Output Sample

Below is one sample output of running the program. **Note that this sample is NOT a comprehensive test.** You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in **bold**. (Note: When you actually run your program no bold or italics should appear at all. These are simply used in this description for clarity's sake.)

Sample Input File (ucfidol2.txt)

```
3
2 4 10
85 90 100 98
78 82 47 77
3 5 12
85 90 100 98 76
78 82 47 77 75
94 6 97 98 99
2 3 15
80 20 50
78 22 53
```

Sample Output

```
UCF IDOL SHOW #1
Contestant #1, you win with an average score of 87.50.
The following judges' scores were thrown out: 3 4


UCF IDOL SHOW #2
Contestant #3, you win with an average score of 97.00.
The following judges' scores were thrown out: 2 3


UCF IDOL SHOW #3
Contestant #2, you win with an average score of 51.00.
All judges' scores were used.
```

Deliverables

One source file: *ucfidol2.cpp* for your solution to the given problem submitted over WebCourses.

Restrictions

Although you may use other compilers, your program must compile and run using Code:Blocks using the C++ compiler. Your programs should include a header comment with the following information: your name, course number, section number, assignment title, and date. Include comments throughout your code describing the major steps in solving the problem. ***Also, you will be graded based on your use of functions.***