

Outline: Your program will process a set of data that contains test scores, five per person.

For example, here is an input with three sets of five grades:

```
99 87 92 86 100
89 76 91 90 88
78 88 98 100 98
```

You will read in the data in `main()`. Then, you will call functions to:

1. validate each score
2. compute the average of each valid set of five scores

Details (a description of each function and then the main function):

0. Write a function **introduction** which will not receive any parameters (and will not return an answer). This function will print a description of the program.

1. Write a function **validate** which will receive an integer representing a test score. The function will determine whether or not the score is valid. A valid score is in the range 0 - 100.

If the score is invalid, the function will print it, with a message explaining why it is invalid-- too big or too small. The function will return a signal saying this is an invalid score. (A `bool` return type is perfect for the answer to be returned by the function.)

2. Write a function **average** which will accept five numbers as parameters and return the average of the five numbers.

Main Function:

a. The main function will call the function **introduction** one time, to start things off.

After this the main function will **loop** to:

b. read in five numbers

c. validate each number

d. calculate the average of the set of grades.

The function **validate** is called on each number to determine if the number is valid. The main function will print whether the set of input is valid (and add to counters).

If the group is valid, the main function will send the numbers another function called **average**. (There will be one call to average.) The main function will print the average of the five scores.

If the set is invalid, you will not calculate the average of an invalid data set! However, the program should continue reading the next set.

The main function will continue this process for the entire set of test scores. You decide how to determine the end of the entire data set. Be sure to explain your choice to the user and in a comment.

When the main function runs out of groups (you must decide when the main function has run out of data), the main function will print the final values of three counters it has been keeping track of: the total number of groups processed, how many groups were valid, and how many groups were invalid.

Use **three separate counters**. Do **NOT** compute one from the others.

DATA: You will be judged on the quality of your data. Include at least 20 groups of various types. Include at least 5 invalid groups (you must spread them through the entire set of data), some with the first grade invalid, some with the second or third invalid, and some with more than one invalid grade.

You are expected to use a data file for the input to the program; you also should use a file for output. Be sure to submit a copy of both the input and output files.

STYLE: Each function should have a good comment explaining its role in the program and what parameter(s) it will receive. Each block should be indented.

OUTPUT: Make sure your output is clean, neat and easy to read.