

[illegible]

Error checking: Your program should print the same output as mine on standard error if too few arguments are given, or if they are not numbers. It should exit silently if any of the parameters is less than or equal to zero, and if the ASCII value of the starting character plus the cycle size is greater than 127.

src/moonglow.cpp

Your sister Sheri is an Philosophy professor. She recently gave a midterm exam to 500 students in her intro class, and it was graded by her PhD student, Moonglow (philosophy students have names like these), who put each student's test scores into a text file with a rather byzantine format. Unfortunately, before Moonglow could convert those files to Excel for Sheri, his wife Sunshine left him, got into their VW van with their child Starrlite and drove home to the commune in northern California where they grew up. That should not surprise you. Moonglow of course followed her, and since the commune is off the grid (duh), Sheri is stuck with 500 text files that she doesn't know how to process.

Fortunately, you're in CS202, and Sheri has offered you a quarter of her grant money to convert those files into a more palatable format. \$25 isn't a huge amount of money, but you love your sister, and this shouldn't take you long, so you agree. You are to write the program **src/moonglow.cpp** which reads a file in Moonglow's format on standard input, and prints a very simple output.

Here's Moonglow's format. The text file is composed of words. If a word is a number, then that is a student's score on a question, so you add it to the student's exam score. If the word is not a number, but is the word "NAME", then the next word is the student's name (Moonglow only uses first names -- last names are corporate and impersonal). If the word is "AVERAGE", then you start reading numbers until you read a word that is not a number (or is the end of the file). You average all of those numbers and add that to the score. Since Moonglow is a little scatterbrained, sometimes a number does not follow "AVERAGE." In that case, you ignore the "AVERAGE".

When you are done reading the file, your program should print the student's name, a space, and the student's score. Just use **cout** for this -- nothing exciting.

Error Checking

You don't have to check for errors, and you may assume that the input is in the correct format. That would cost Sheri more money. However, this is a pretty flexible format. The only things that you need to assume are:

- The input will specify exactly one student's name.
- That student's name won't be "AVERAGE" or "NAME."

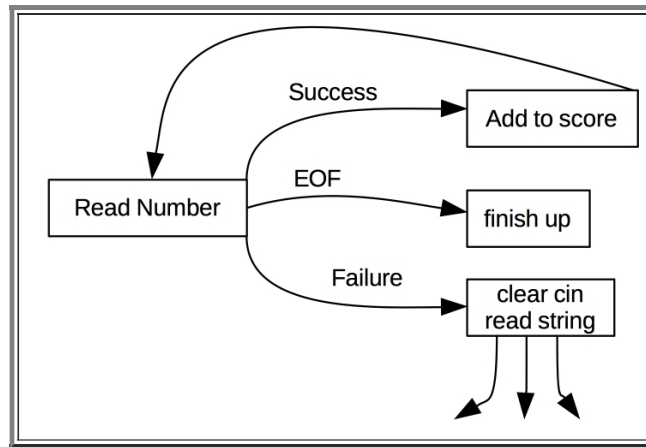
Here are some examples:

File	Contents	Correct output
data/test-1.txt	NAME Fred	Fred 0
data/test-2.txt	NAME Dontonio 8.6 16.2	Dontonio 24.8
data/test-3.txt	I miss Starrlight!! 15 NAME Frank	Frank 15
data/test-4.txt	AVERAGE 10 15 20 NAME Luther 10 Starrlite!!	Luther 25
data/test-5.txt	NAME Baby-Daisy AVERAGE 3 4 5 6 AVERAGE 7 8 9 Where's Starrlite!!	Baby-Daisy 12.5

data/test-6.txt	<pre> Starrlite AVERAGE Starrlite!!! NAME Starrlite AVERAGE 55 Starrlite!!! Starrlite AVERAGE Starrlite 5 6 7 </pre>	Starrlite 73
---------------------------------	--	--------------

Some comments on the examples. First, the file doesn't have to have any scores, like [data/test-1.txt](#). Second, you should use doubles and not integers. Third, the "NAME" can appear anywhere, and can be on a different line than the name, as in [data/test-3.txt](#). Last, the "AVERAGE" can have zero numbers after it, as it does twice in [data/test-6.txt](#). Thus, in the file, the score is 55, from the second "AVERAGE", plus 5, 6 and 7, which equals 73.

A suggestion: This lab frustrates students, because you have to do your operations in a certain order. I suggest that you draw a block diagram to help you figure out the order of operations. That will make your programming much easier. Here's a start to the block diagram:



The Gradescript

The grading programs are in `/home/jplank/cs202/Labs/Lab1/gradescript` and `/home/jplank/cs202/Labs/Lab1/gradeall`

The first 30 tests are for **checkerboard**. The remaining 70 are for **moonglow**.