**Before You Begin**

Thank you for putting your time into taking this challenge and exploring the opportunity with BookBub. **Make your best effort to package and structure your program to allow for understandability and future maintenance**—that is, write this code with the same quality as a feature you'd be shipping to bookbub.com. There's no major gotchas or brain-teasers involved here, we're primarily interested in getting a sense for your practical ability to write code and solve an almost real-world problem. We don’t expect you to spend more than a small handful of hours on this, so feel free to timebox yourself to two or three hours. If you run out of time, you can document in the readme what you would have done with more time.

**The Challenge**

We want to build a tool to help BookBub’s editorial team quickly go through a list of books and identify the correct genre for each book. The approach we’re taking is to analyze the description of each book and calculate a score based on the presence of certain genre-specific keywords.

**Calculating the Score**

For each genre, we have several keywords that we expect to be present in the book description for that genre. We have assigned a point value to each keyword because some keywords provide stronger signals than others. The genre-fit score is calculated as:

total num keyword matches \* avg point value of the **unique** matching keywords

Example Book:

{ “title”: “Hunger Games”, “description”: “In a not-too-distant future, the United States of America has collapsed, weakened by drought, fire, famine, and war, to be replaced by Panem, a country divided into the Capitol and 12 districts. Each year, two young representatives from each district are selected by lottery to participate in The Hunger Games. Part entertainment, part brutal intimidation of the subjugated districts, the televised games are broadcast throughout Panem as the 24 participants are forced to eliminate their competitors, literally, with all citizens required to watch. When 16-year-old Katniss's young sister, Prim, is selected as the mining district's female representative, Katniss volunteers to take her place. She and her male counterpart, Peeta, the son of the town baker who seems to have all the fighting skills of a lump of bread dough, will be pitted against bigger, stronger representatives who have trained for this their whole lives. Collins's characters are completely realistic and sympathetic as they fight and form alliances and friendships in the face of overwhelming odds; the plot is tense, dramatic, and engrossing. This book will definitely resonate with the generation raised on reality shows like 'Survivor' and 'American Gladiator.' Book one of a planned trilogy.” }

Example Keywords CSV:

**Genre, Keyword, Points**

action, fast paced, 7

action, distant future, 4

action, fight, 6

mystery, murder, 5

mystery, death, 8

mystery, explosive, 4

sci-fi, future, 8

sci-fi, spaceship, 9

In the example above, the *action* score for Hunger Games is 15. Keywords “distant future” and “fight” both match and their average points are 5. With 3 total keyword matches, the *action* score is 15, with 3 \* 5 = 15. The *sci-fi* score is just 8, since “future” is the only *sci-fi* match. The *mystery* score is 0 since there are no keyword matches.

## Your Challenge

Write a command-line program that takes in two arguments:

1. A json file containing a list of books ([sample json file](https://drive.google.com/file/d/0BwH8Eql0y7ECaGNGYThjNUpsZjg/view?usp=sharing))
2. A CSV with genre/keyword/value rows ([sample csv file](https://drive.google.com/file/d/0BwH8Eql0y7ECU2JVRmt6WGJiM00/view?usp=sharing)).

Your program should print out the book titles alphabetically, and underneath each book title should be the three highest scoring genres with a score greater than 0 (including the score for that genre).

**When run against the example files linked above, your program should produce the following output:**

Hunger Games

action, 15

sci-fi, 8

Infinite Jest

literary fiction, 12

You can write your program in any language you like as long as we can run it on Linux.

Once you’ve completed your exercise, please send us a zip of your source code along with a README.txt explaining:

* Any steps necessary to run your program
* Any interesting trade-offs or edge cases you ran into
* Approximately how long you spent (this is not timed, but it’s helpful for us)

**Notes:**

The purpose of this exercise is for you to show us your ability to write clean, non-trivial code. We realize the score calculation is too simplistic to be very good, but for the purposes of this exercise that’s okay.

The genre keywords can be up to two words separated by a space. For example, if the keyword is “distant future” and the description only contains “distant past and future” then there is not a match.