Problem Description

We are developing a system to rank companies around the world based on theirs customer's opinions. All companies interested in being part of this rank allow us to submit surveys to their customers and collect the answers. Your task is to process the answers and generate the rank.

See the details:

* All company's customers receive the same survey, with the same set of questions.
* Every question has an unique id.
* Every question is "multiple choice" and allows 5 possible answers: 0, 1, 2, 3 and 4.
* Answers 0 or 1 are considered "favorable", 2 is "neutral" and 3 or 4 are "unfavorable".
* Question sample send to a hotel customer:

I felt that the hotel's staff has been helpful.

( ) (X) ( ) ( ) ( )

Very helpful Staff wasn't helpful at all

* In the sample above the customer answered answer 1, considered "favorable". It means he or she aproves the hotel staff.
* The customers answer the surveys on a legacy web site we own.
* The web site back-end generates text files with the answers.
* One file is generated for each company.
* Due legacy system bugs we may have answers different from 0, 1, 2, 3 and 4. Such answers must be discarded.
* Our new system will grab the files, process them and generates the rank.

Take a look at this file sample:

Foobank
4569 2
4567 0
4568 1
4570 5

* The first line is the company name and the following lines are the question id followed by its answer.
* Note that question 4570 has an invalid answer and it must be discarded.
* Your program must take a list of files as parameter (consider full path) and output the rank. See examples of call in java and python:

java rank /tmp/hall_a.i._ltd._iCiSXO.txt /tmp/it_services_iqMO2z.txt /tmp/logistic_company_AbOzZl.txt /tmp/mybank_zqweSt.txt /tmp/pineapple_store_abXKXL.txt

python rank.py /tmp/hall_a.i._ltd._iCiSXO.txt /tmp/it_services_iqMO2z.txt /tmp/logistic_company_AbOzZl.txt /tmp/mybank_zqweSt.txt /tmp/pineapple_store_abXKXL.txt

* The rank is a report with the following format:

Summary by companies:

IT Services
4568: 41% fav, 19% neutral, 40% unfav
4570: 40% fav, 18% neutral, 40% unfav
4569: 39% fav, 19% neutral, 40% unfav
4571: 38% fav, 19% neutral, 41% unfav
4567: 37% fav, 20% neutral, 41% unfav

MyBank
4568: 66% fav, 0% neutral, 33% unfav
4571: 50% fav, 0% neutral, 50% unfav
4570: 50% fav, 0% neutral, 50% unfav
4569: 0% fav, 100% neutral, 0% unfav
4567: 0% fav, 100% neutral, 0% unfav

Logistic Company
4569: 41% fav, 23% neutral, 34% unfav
4568: 35% fav, 20% neutral, 43% unfav
4571: 37% fav, 24% neutral, 38% unfav
4567: 32% fav, 21% neutral, 45% unfav
4570: 34% fav, 21% neutral, 44% unfav

Fav answers by questions:

4567: IT Services 37% fav, Logistic Company 32% fav, MyBank 0% fav
4568: MyBank 66% fav, IT Services 41% fav, Logistic Company 35% fav
4569: Logistic Company 41% fav, IT Services 39% fav, MyBank 0% fav
4570: MyBank 50% fav, IT Services 40% fav, Logistic Company 34% fav
4571: MyBank 50% fav, IT Services 38% fav, Logistic Company 37% fav

Valid answers:

IT Services: 3256
Logistic Company: 4563
MyBank: 1023

Invalid answers:

IT Services: 0
Logistic Company: 35
MyBank: 17
* On "Summary by companies" the questions must be presented in descending order of favorable answers %.

* On "Fav answers by questions" the questions must be ordered by id and the companies ordered in descending order of fav answers %.
* On "Valid answers" you must show the valid answers count. The data must be presented in alphabetic order.
* On "Invalid answers" you must show the invalid (discarded) answers count. The data must be presented in alphabetic order.
* Remember that files may contain invalid answers. You must discard those answers and do not use them on calculations.
* You can output the rank on the terminal, no need to generate a file.
* Small rounding errors are acceptable.

About your delivery:

* Use the language of your choice.
* You must deliver a zip file with your source code and a readme file with the following instructions:
    * Environment requirements like JDK version, Python version, virtualenv, constraints, etc.
    * How to build and run your program.
* If you are a .Net developer:
  * Consider using .NET Framework version 4.0. If this is a problem let us know in advance, we will find a way to evaluate your code.
  * We will use a Linux host and MonoDevelop to run and evaluate your code. Feel free to code with Visual Studio, MonoDevelop can read and execute those projects.
* If you decide to use 3rd party libraries pack them into the zip file. Do not use build tools like maven, gradle, etc.
* You can use the sample files attached to test your program.
* Do not pack executables, jar files, etc.. We will compile and execute your code.
* We encourage you to write unit tests and send them with the solution.
* Have fun!!