Tester Matching

**Background**

A major feature of Applause is our tester-matching algorithm. We are able to drilldown from a community of over 175k testers with multiple devices, to 10-20 testers that best meets a customer needs. As you can imagine, this is a complicated algorithm that takes multiple dimensions into account.

Your goal will be to write a simpler matching algorithm that takes two matching criteria (country and device) and presents a sorted list of results. The sort order is based on what the system is suggesting a user should select (more details below).

**Data Set Provided**

* testers.csv: CSV list of testers. Each row corresponds to a tester.
* devices.csv: CSV list of all available devices. Each row corresponds to a mobile device. This is all the possible devices a tester can have.
* tester\_device.csv: CSV list of tester to device. These are the devices a tester has. Each row corresponds to a tester and a device.
* bugs.csv: CSV list of all the bugs filed by a tester. Each row corresponds to a bug filed by a tester and contains a tester and the device the bug was reported on.

# Assignment

Write an application that will match testers based on a user search criteria. The search results should be ranked in order of experience. Experience is measured by the amount of bugs a tester filed for a given device. You can use any third party libraries, but be prepared to discuss why you decided to use it. You may use any technology, but please have a browser based UI for the search and results.

**Search Criteria**

* Country: values should be collected from "tester.csv" and should also have an option for "ALL". User can select one or more countries. Multiple selections are treated as "OR".
* Device: values should be collected from "devices.csv" and should also have an option for "ALL". User can select one or more devices. Multiple selections are treated as "OR".

**Note about walk:**

The walkthrough examples are not using the data set provided and the output is to help explain how we derived to the end answer.

**Walk Through Example 1**

Search Criteria: Country="ALL" and Device="iPhone 4"

Matches: 2 testers (User1 and User2).

User1 filed 4 bugs for iPhone 4.

- 4 bugs filed for devices in search

User2 filed 10 bugs for iPhone 4.

- 10 bugs filed for devices in search

Results: User2, User1

**Walk Through Example 2**

Search Criteria: Country="ALL" and Device="iPhone 4" or "iPhone 5"

Matches: 2 testers (User1 and User2).

User1 filed 4 bugs for iPhone 4 and 20 bugs for iPhone 5.

- 24 bugs filed for devices in search

User2 filed 10 bugs for iPhone 4.

- 10 bugs filed for devices in search

Results: User1, User2

**Walk Through Example 3**

Search Criteria: Country="ALL" and Device="iPhone 6"

Matches: 2 testers (User3 and User4).

User3 filed 4 bugs for iPhone 6

- 4 bugs filed for devices in search

User4 filed 0 bugs

- 0 bugs filed for devices in search

Results: User3, User4

**UI Output**

How you output the results are up to you, there are no design requirements.

# Submission

Compress entire source code tree, include third-party libraries and instructions how to run or access the running example. Bonus points for hosting the solution UI somewhere (discretely).

# Presentation

Brad will get in contact with you on scheduling time for you to present and walk through your solution. During the presentation, there will be a couple engineers that will ask various questions about the code.

# Questions

If you have questions or problems, feel free to email us to ask. We’re happy to provide input on whatever you like. For questions about the assignment, make sure to include these people in your email:

* Michael – mlubavin@applause.com
* George – gbabey@applause.com