Coding Challenge

If you are reading this, that means that Ocrolus is really interested in your engineering abilities. Congratulations!

This is meant to be a small programming exercise to gauge your programming style and skills.

Included are two files that you will write a program to process. Java or Python is preferred. After you've engineered a solution, email us the source code and result of running the input file (queries.txt) through your program.

Your solution should be documented and tested to production standard. Your code will be assessed on design, correctness, code quality and maintainability, and quality of unit testing. This is not a timed coding test, you will not be graded on how quickly you submit your solution.

Feel free to use any open-source libraries as needed.

If you have any questions, please reach out to us at careers@ocrolus.com

The problem

You will be creating a phone book search system.

You're tasked with taking entries of personal information for a phone book in multiple formats and storing it in a appropriate data structure of your choice to allow for searching for a phone number for a person using their last name.

The input file (phone_dataset.csv) has records in the below three formats, each line containing their last name, first name, state and phone number.

```
Lastname, Firstname, New York, (917) 958-1191
Firstname Lastname, 9179581191, New York
Firstname, Lastname, (917) 358-1291, California
```

All phone numbers are US phone numbers (10 digits).

If the line is invalid you can ignore it and it should not interfere with processing of subsequent lines.

Your program should load the phone_dataset.csv which has the phone book data, take in search query input from the queries.csv, and output results to output.csv.

The format of the queries.txt is a list of last names (one per line), for example

```
Smith
Doe
Tyson
```

For each line in the queries.txt file, write results to output.txt file in the below format. Note that there can be multiple matches for a given lastname. The results should be sorted by first name.

```
Matches for: Smith
Result 1: Lastname, Firstname, State, Phone
Result 2: Lastname, Firstname, State, Phone
Matches for: Doe
Result 1: Lastname, Firstname, State, Phone
…
Result N: Lastname, Firstname, State, Phone
Matches for: Tyson
No results found
```

See sample run on next page

Sample

You are provided - phone_dataset.csv

```
Doe, John, New York, (917) 958-1191
Doe, John, California, (212) 234-1191
John Smith, 9179581191, New York
Bill, Gates, (917) 358-1291, New York
Doe, John, Florida, (919) 234-1192
```

You are provided - query.txt

```
Doe
Smith
GAtes
Abc
```

You should produce below output.txt

```
Matches for: Doe
Result 1: Doe, John, New York, (917) 958-1191
Result 2: Doe, John, California, (212) 234-1191
Result 3: Doe, John, Florida, (919) 234-1192
Matches for: Smith
Result 1: Smith, John, New York, (917) 958-1191
Matches for: GAtes
Result 1: Gates, Bill, New York, (212) 333-1191
Matches for: Abc
No results found
```