ENNOVA
Engineering Innovation.

Programming Problem - Search

Introduction

Develop an application using Ruby to solve the problem described below. The objective of the problem is to allow the candidate to **demonstrate their skill and experience** in aspects of the development process including domain modelling, object-oriented design and analysis, use of language constructs and features, testing (unit or otherwise), etc...

The problem is provided with sample data to be used for testing and the candidate should be able to demonstrate that their solution using the supplied data in the form of a unit test or a simple user interface. User interface design is not the main focus of the problem.

Resulting solution files should be returned by email at least 48 hours before your interview.You will be required to discuss your solution at the interview.

Problem Background

COMPANY XXX is particularly interested in improving the volume and quality of traffic to a our public facing web sites from search engines. Many of these search engines are quite sophisticated, using advanced algorithms and parallel searching techniques to provide fast, accurate responses.This problem is however, somewhat simpler.

A group of web pages has been classified by associating a list of keywords, given in decreasing order of relevance, with each page (i.e., the order of keywords is from the most specific keyword to the least specific). For example, on the TopGear web site a page on reviews of Ford cars may have the keywords: Ford, Car, Review in that order; the most relevant keyword is Ford. Queries also include a list of keywords, again from most to least relevant. For example, in a query consisting of the keyword Ford followed by the keyword Car, Ford is more important than Car.

In this problem you are to determine the top five (or fewer) pages that match each of an arbitrary number of queries.To determine the strength of the relationship between a query and a web page, assume the keywords for each page and each query are assigned integer weights, in descending order, starting with N, where N is the maximum number of keywords allowed for a web page and query.The strength of the relationship is the sum of the products of the weights associated with each keyword that appears both in the web page list and the query list.

For example, assume the following web pages and keyword lists:

```
Page 1: Ford, Car, Review
Page 2:Toyota, Car
Page 3: Car, Ford
```

For N equal 8, a query with keywords Ford and Car in that order yields the following strength ratings.

```
Page 1: (8×8 + 7×7) = 113
Page 2: (7×7) = 49
Page 3: (7×8) = 56.
```

Similarly, a query with keywords Ford and Review yields the following strength ratings.

```
Page 1: (8×8 + 7×6) = 106
Page 2: = 0
Page 3: (7×8) + (8x7) = 112
```

Input

Input data consist of one line for each web page and query.A line consists of a code letter followed by a list of keywords. Code letters P and Q denote a page and a query. Code letters and keywords are separated by at least one space. Ps and Qs may occur in any order.

Pages are added sequentially starting with page one. The case of characters in the keywords is not significant. Each query also has of a list of between one and 8 keywords.Again, case being insignificant for keywords. Number the queries sequentially starting with one.

Output

For each query, identify the 5 (or fewer) pages stored that are the most relevant to the query. Print a single line containing the query identifier, a colon, and the page identifiers of the five most relevant pages in the decreasing order of relevance. Page identifiers consist of the letter "P" followed by the page number. Query identifiers consist of the letter "Q" followed by the query number. If several pages have the same relevance, list them by increasing page number. Do not list pages that have no relationship (zero strength), even if fewer than five pages are identified.

Additional Considerations

Although the search strength algorithm described here is quite simple, developers should make provision for substituting a more complex method in the future and consider the impact of nested pages.

Sample Input

```
P Ford Car Review
P Review Car
P Review Ford
P Toyota Car
P Honda Car
P Car
Q Ford
Q Car
Q Review
Q Ford Review
Q Ford Car
Q cooking French
```

Output for the Sample Input

```
Q1: P1 P3
Q2: P6 P1 P2 P4 P5
Q3: P2 P3 P1
Q4: P3 P1 P2
Q5: P1 P3 P6 P2 P4
Q6:
```