

Countries

Problem ID: a14p04countries

Write a program that reads from a text file a list of countries in the world, and then lists all country names of a given length.

The name of the text file is supplied by the user, so the program should begin by collecting the file name. In case the given file name is invalid, the program should report the fact, and demand a new one, repeatedly until a valid file name is entered.

Once a valid file has been given, the program should allow the user to repeatedly supply an integer, and each time the program should respond by listing all countries the length of whose names is equal to that integer.

Rather than having to go through the entire list each time the user queries for a specific length, the program should process the country list before starting to take length queries from the user.

That's where the subject of today's class comes in, dictionaries. As the program reads the contents of the given file, it should check the length of each country name, and organize the countries into a dictionary, where countries with equally long names are grouped together.

So the program should create a dictionary where the key is an integer representing the length of a country name and the value is a list of country names of that length. For example, an entry in the dictionary might look like this:

```
• {16 : ["Marshall Islands", "Papua New Guinea"]}
```

Note: the file `countries.txt` that is used in the samples is attached for your convenience.

Input

The first line of the input is taken to be the name of a file, and possibly more lines as well, as many as needed until a valid file name is provided. After that, the rest of the input will come in pairs, an integer on one line, denoting a query, followed by a line containing a string indicating whether there are more queries to come, until the string will be the single letter `n`. Each line in the input file will contain one country name.

Formal input specifications

Formally, the input will consist of $m + 2n$ lines, l_1, \dots, l_{m+2n} , where m denotes the number of file name attempts, with $1 \leq m$, and n denotes the number of queries, with $1 \leq n$.

For $i < m$, line l_i will contain an invalid file name attempt a_i , and line l_m itself will contain a valid file name a_m .

For $j \in \{1, \dots, n\}$, line l_{m+2j-1} will contain an integer q_j , and line l_{m+2j} will contain a string r_j , denoting a reply to the question whether more queries will follow. The reply r_n will consist of the string `n`, indicating that no more queries are forthcoming. All other replies r_j for $j < n$ will be something other than `n`.

So line l_{m+1} will contain a query q_1 , line l_{m+2} a reply r_1 , line l_{m+3} a query q_2 and so on, until the reply r_n will be `n`.

The input file will consist of k lines, with $0 \leq k$, and each line will contain one country name, c_i , for $i \in \{1, \dots, k\}$.

Test case restrictions

In the tests, m and n will be restricted to $m \leq 10$ and $n \leq 1\,000\,000$. The length of each file name attempt a_i will be restricted to $1 \leq |a_i| \leq 100$. Each query q_j will be restricted to $1 \leq q_i \leq 50$, and each reply r_j will be either `y` or `n`, with r_j being `y` for all $j < n$, and the last reply r_n being `n`.

As for the contents of a valid input file, k will be restricted to $0 \leq k \leq 1\,000$, and the length of each country name c_i will be restricted to $1 \leq |c_i| \leq 40$. The countries will appear in alphabetical order in the file.

Output

For each invalid file name attempt a_i , with $i < m$, the program should report that the file was not found. And then, for each query q_j , with $j \in \{1, \dots, n\}$, the program should display all countries whose names have length q_j , on a single line separated by commas, or state that no such country was found.

Formal output specifications

So, the output should consist of $m - 1 + n$ lines in total, call them $\lambda_1, \dots, \lambda_{m-1+n}$. For $i < m$, line λ_i should contain the string:

- File $\{a_i\}$ not found!

And then, for each query q_j , with $j \in \{1, \dots, n\}$, line λ_{m-1+j} should contain all country names c_i satisfying $|c_i| = q_j$, separated by commas as well as a space, as long as there is at least one such country. For example, if there are 3 country names $c_{i_1}, c_{i_2}, c_{i_3}$ of length q_j , then line λ_{m-1+j} should be:

- $\{c_{i_1}\}, \{c_{i_2}\}, \{c_{i_3}\}$

The country names should be listed in the order they appear in the file.

If there are no countries of the given length q_j , then line λ_{m-1+j} should contain:

- No country name of length $\{q_j\}$ exists.

Sample Input 1

```
Atlantis
El Dorado
countries.txt
4
Y
16
n
```

Sample Output 1

```
File Atlantis not found!
File El Dorado not found!
Chad, Cuba, Fiji, Iran, Iraq, Laos, Mali, Oman, Peru, Togo
Marshall Islands, Papua New Guinea, St Kitts & Nevis
```

Sample Input 2

```
countries.txt
6
n
```

Sample Output 2

```
Angola, Belize, Bhutan, Brazil, Brunei, Canada, Cyprus, France, Gambia, Greece, Guinea, G
```

Sample Input 3

```
countries.txt
2
n
```

Sample Output 3

```
No country name of length 2 exists.
```