Document Retrieval
Problem ID: p09documentretrieval

Warning: This is a challenging project, probably the most difficult project that you have worked on so far in the course. Thus, it is even more important now than before to start working on this project as early as possible.

Background: Document retrieval is the task of finding documents that meet the search criteria input by a user. The most well-known example is web search, where a user types in a set of keywords and the search engine finds web pages that are relevant to the search query. True document retrieval can be quite difficult, as it needs to take into account many different factors. In this project you will implement a very simple document retrieval engine.

Specification: In this project, document collections are stored in text files. At the end of each article in a collection there is a line that contains only the string `"<END OF DOCUMENT>"`. Your program prompts the user for the name of a text file containing a document collection, reads in the documents from the file and stores the content as a list of strings, with one (long) string for each document. If the document collection file is not found, the program should quit without any output.

In order to look up search terms, the program needs to keep track of which words appear in each document. You should use a **dictionary** for this purpose. Each entry in your dictionary should have a lower case word as the **key** and the **value** should be the **set** of document numbers in which this words appears. Punctuations at the start and end of a word should be removed.

The program allows a user to perform three actions:

1. **Search**: If this option is selected, the user is prompted for a search string. The program then prints out the number of the documents (in ascending order) in the collection containing every individual words/terms (case-insensitive) in the search string. The order of the words in the search string is irrelevant. If no documents in the collection contain every term in the search string, the program prints the message `"No match"`.

2. **Print**: If this option is selected, the user is prompted for a number for a document. The program then prints out the entire content of the given document. If the document corresponding to the number is not found, the program prints `"No match"`.

3. **Quit**: If the user inputs an action which is neither 1 nor 2, the program quits.

Your program should continue to prompt until the user chooses to quit.

Example: An example document collection file (`example.txt`) looks like this:

```
This is an example document collection file.
It contains three documents.
This is text from the first document.
<END OF DOCUMENT>
This is text from the second document. Each document is
of variable length.
<END OF DOCUMENT>
And now we are in the third document, which is indeed
the "largest" one
measured in the length of the
text that appears in the document.
<END OF DOCUMENT>
```

Input

The first line of the input contains the name of a document collection file containing text from one or more documents. Thereafter, iteratively, in line $2i$, where $1 \leq i \leq 5$, the input is one of the following:

- The string `"search"`: In this case, line $2i + 1$ contains a sequence of one to three strings (search terms) separated with a white space.
- The string `"print"`: In this case, line $2i + 1$ contains a single integer denoting the document to be printed.
- The string `"quit"`: In this case, no more input occurs.

Output

When the document collection file cannot be opened, no output is generated. Otherwise, in each iteration, the output is either:

- A response to a search query which is either:
  - The string `"Documents matching search: "`, followed by a sequence of document numbers (in ascending order) in which the entered search terms were found,
  - or the string `"No match"` if no documents match the search string.
- A response to a print action which is either:
  - The string `"Document number {i}:"`, where i is the number of the document to be printed out, followed by (in the next line) the text of the file number i,
  - or the string `"No match"` if no document in the collection corresponds to the entered number i.

Scoring

Each of the 20 hidden test cases is also worth 5 points. The test cases are accompanied with hints, which you can see if you get them wrong, describing what is being tested.

Sample Input 1

```
example.txt
print
2
search
largest
print
3
print
1
search
document
quit
```

Sample Output 1

```
Document #2:
This is text from the second document. Each document is
of variable length.
Documents matching search: 3
Document #3:
And now we are in the third document, which is indeed
the "largest" one
measured in the length of the
text that appears in the document.
Document #1:
This is an example document collection file.
It contains three documents.
This is text from the first document.
Documents matching search: 1 2 3
```

Sample Input 2

```
example.txt
print
4
search
document from
search
the text document
search
programming
print
3
quit
```

Sample Output 2

```
No match
Documents matching search: 1 2
Documents matching search: 1 2 3
No match
Document #3:
And now we are in the third document, which is indeed
the "largest" one
measured in the length of the
text that appears in the document.
```