Skilaverkefni 8 / Project 8
Document retrieval

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 key words and the search engine finds web pages that are relevant to their 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.

Program specification

In this project, document collections are stored in text files. Each article in a collection starts with a line that contains only the string <NEW 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 one long string) of each document in a **list**. The first document found is in position 0 of the list, the second document in position 1, etc. If the document collection file input by the user is not found, the program prints "Documents not found." and quits.

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 word as the **key** and the word's **value** as the **set** of documents that this word appears in. This arrangement allows you to look up a keyword in the dictionary and immediately get all the documents that it appears in, making it easy to figure out documents that might meet a search query.

The program allows a user to perform three actions:

1. **Search documents**: The user inputs a search string. The program then prints out the number of the documents in the collection containing every individual words/terms in the search string. If no

documents in the collection contain every term input by the user, the program prints the message "No match.".

2. **Print document**: The user inputs a number for a document. The program the prints out the entire content of the given document.
3. **Quit the program**. 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 input/output:

```
Document collection: ap_docs.txt

What would you like to do?
1. Search Documents
2. Print Document
3. Quit Program
> 1
Enter search words: stock prices
Documents that fit search: 16 2 221 222

What would you like to do?
1. Search Documents
2. Print Document
3. Quit Program
> 2
Enter document number: 2
Document #2
The stock market closed out its worst week so
far this year, as prices fell for the second
straight session.
The Dow Jones average of 30 industrials dropped
44.92 points
Friday to 1,978.95, finishing the week with a net
loss of 108.42.
That marked the average's biggest weekly decline
since it
dropped 143.74 points last Nov. 30-Dec. 4.
...
```

```
What would you like to do?
1. Search Documents
2. Print Document
3. Quit Program
> 3
Exiting program.
```

Further instructions:

1. Search queries should not be case sensitive, i.e. searching for “Stocks” should give all documents that contain ‘stocks’, ‘STOCKS’, etc.
2. You should remove punctuation from the start and end of a word as well. If the string “stock,” appears in the document, this should be counted as an instance of the word “stock” (without the comma).
3. Begin small. We have provided a tiny test file for development: “ap_docs2.txt”. This file has three two-line documents.
4. Both ap_docs.txt and ap_docs2.txt are available on the github repo, in the projects/hw8 directory.