

CSCE100 Introduction to Informatics
Fall 2018

Programming Assignment 3: Dictionary Googler

Points: 100 points. Assignment Date: October 01, 2019 Due Date: October 10, 2019

Objectives

1. To familiarize with writing and running Python programs and the Python environment
2. To familiarize with the use of loops (e.g., the for and while loops)
3. To familiarize with data structures, particularly arrays/lists
4. To familiarize with file input/output in Python
5. To be exposed to the use of built-in functions
6. To be exposed to the use of built-in modules or packages (e.g., import csv)
7. To familiarize with the use of online documentations on Python

Problem

The problem is to build a search engine of words in a dictionary that allows users to enter several parameters for their search query. Search and retrieval of data are key steps in many informatics-related processes. Thus, this assignment gives you a sense of how search and retrieval can be achieved.

Write a program that will read in all the values from a given text file (that will be provided to you) and store them in arrays/lists. After that, the program should prompt the user for four parameters that configure his or her search: (1) the start letter of the words to be retrieved, (2) the last or ending letter of the words to be retrieved, (3) the minimum number of letters in the words to be retrieved, and (4) the maximum number of letters in the words to be retrieved. Thus, if the user enters 'a', 'b', '5', and '9', then the program should return all words found in the dictionary that starts with an 'a', ends with a 'b', has at least 5 letters and at most 9 letters. One example of such a word is "absorb".

An additional twist is that instead of printing out the retrieved words to the screen, the program will write (1) the four search parameters, and (2) the list of retrieved words into an external output file.

Here are some additional requirements and tips:

- You are required to use the given "dictionary.txt" file and read in all the values from it. (10 points) (*Hint:* To read in all the values from a text file you can use a while loop to read row by row. Refer to class notes.)
- You are required to use an array or list to hold the words read in from "dictionary.txt". (5 points)

- You are required to use an array or list to hold the retrieved words as a result of performing search on the dictionary words. (5 points)
- Your program is required to perform some error checking on the search parameters
 - The letters entered must one of the 26 alphabet letters. (5 points)
 - The numbers entered must be non-negative, and that the minimum number cannot be greater than the maximum number. (5 points)
- You will need to remove whitespaces or newline characters at the beginning or end of a string (i.e., a word in this case). (*Hint*: You can use the “.strip()” function that Python has built for us to use. The “.strip()” function returns the copy of the string after removing the beginning and trailing characters.) (5 points) For Example:

```
myWord = "hello\n"
myWord = myWord.strip() # call strip() and overwrite "myWord"
```

Output: hello

- Your program is required to generate the output of your analyses to an “output.txt” file. (10 points) (*Hint*: See “Output.txt” below for an example.)
- You must document your program (see <https://devguide.python.org/documenting/>).
 - Name, Date, Affiliation, a description of the program, what inputs does it need, what outputs does it generate. (5 points)
 - Inline comments in the program. (5 points)

Example Input/Output:

Welcome to the Dictionary Googler program!!!

Enter the first letter: a

Enter the last letter: b

Enter the minimum number of letters in the word: 5

Enter the maximum number of letters in the word: 9

Successfully generated output file Output.txt.

Thank you for using the Dictionary Googler program.

output.txt

Search Parameters:

The starting letter is a

The ending letter is b

Minimum number of letters is 5

Maximum number of letters is 9

Number of words retrieved = 16:

abcoulomb
absorb
acerb
adsorb
adverb
aerobomb
aintab
aldicarb
algenib
antisnob
aplomb
ardeb
arneb
athrob
attrib
aurangzeb

Thank you for using this program!!!
GOODBYE!

Handin

1. The submission deadline for all handins is 11:00 AM October 10, 2019. **Late handins will *not* be accepted or graded.**
2. You are required to handin a screen capture of your “testing session” using your program. (10 points)
3. You are required to handin all program files. (10 points)
4. You are required to handin all input and output files. (5 points)
5. You are required to handin online the above using <http://cse.unl.edu/handin/>

Bonus: 15 points

The above program requirement prompts the user for input. But there is another way in which we can provide the code with the required information without prompting the user. That is, we can invoke the program and supply the search parameters directly by using *command line arguments*. For example,

```
> python myDictionaryGoogler.py a b 4 9
```

For this bonus part, revise your original solution such that if the command line invocation has non-empty arguments, then use them accordingly as search parameters; otherwise, prompt the

user as usual for them to input the four search parameters. Then the program performs the analysis as usual in both cases.

Think About

Now, think about what if we want to modify how the rows are arranged (e.g., adding new attributes, removing some attributes) or how the rows are to be filtered so that only certain subsets are being considered for analyses. Do we need then to create many versions of the input file? Furthermore, think about the output file format. What if our users or customers want something different, e.g., in a different format, or to derive or visualize different types of results from your analyses? How would you then create an output file that is more universally useful? (Hint: Think about databases.)