ÇANKAYA UNIVERSITY

CENG 383 Algorithms - Programming Assignment 2

In this assignment, you will write a program that performs **spell checking** on input strings taken from the user. At start, your program should take a txt file as the command-line argument: a file containing the dictionary.

Your program should read the dictionary, and build a Trie-based index for it, as we discussed in the class. Then it will wait input strings from user. For each input string given, it performs a search for the string on the trie. If it is misspelled, the program should print "misspelled?" and then suggest three **alternatives** (or less than three if there are not enough prefix matching words), with the **longest matching prefixes**. Otherwise, it prints "correct word".

Your implementation should work in **constant time** (or **linear time with respect to number of characters** of the string given as input).

INPUT FORMAT

Your program should read the given **dictionary.txt** file with the format:

```
ABACUS
ABANDON
ABANDONED
ABBREVIATE
ABILITY
ABLE
ABOUND
ABOUT
ABOVE
ABRACADABRA
ABSENCE
ABSENT
…
```

OUTPUT FORMAT

Sample run of the program using the dictionary file given on the left:

```
input -> java spellcheck.java dictionary.txt


input -> able

output -> correct word

input -> abo

output -> misspelled? abound, about, above

input -> absenn

output -> misspelled? absence, absent, abacus
```

Remarks

- The main function in assignment should take 1 command line argument, the name of dictionary file. Then it should wait for a string input from the user.
- You can test your implementation on the given sample dictionary file or using some part of it. Also, you can test on different files to be sure that your program also works for different dictionaries.
- Your implementation **will be tested on different dictionaries having thousands or more words**. Your implementation should work in constant time (or linear time with respect to number of characters of the string given), otherwise you will get no credits.
- For this assignment, you **ARE ALLOWED** to use the codes given in our textbook and/or our lecture slides. You **ARE NOT ALLOWED** to use any codes from somewhere else (e.g., from the internet, other textbooks, other slides ...). You **ARE NOT ALLOWED** to use external libraries including our textbook.

WHAT TO HAND IN

A zip file containing:

- The Java source files (only with **.java extension**) of your program. Don't upload your codes in txt or any other format!
- The Java codes should be **WELL FORMATTED** and **WELL DOCUMENTED** as comments if necessary, as part of your grade will be based on the level of the format and clear comments.

IMPORTANT

IMPORTANT NOTES: Do not start your homework before reading these notes!!!

1. This assignment is due by 23:55 on Sunday, May 26th.

2. You should upload your homework to **WebOnline** before the deadline. No hardcopy submission is needed. You should upload files and any additional files if you wrote additional classes in your solution as a **single zip file**.

3. The standard rules about late homework submissions apply. Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

4. The submissions that do not obey these rules will not be graded.

5. Since your codes will be checked without your observation, you should report everything about your implementation. Add detailed comments to your classes, functions, declarations etc if necessary.

   ```
   //------------------------------------------
   // Summary: Assigns a value to the variable
   //------------------------------------------
   void setVariable(char varName, int varValue) {
           // body of the function
   }
   ```

   - Indentation, indentation, indentation...

   - You are welcome to ask your questions on the **Office Hours** (Mondays 13.30-15:00 L117)

You do not need to prepare a PDF report, just document your code well. You may be asked to explain your code as a face-to-face demo session. The schedule will be announced later in that case.