|  |  |  |
| --- | --- | --- |
| **LAB101Assignment** | **Type:** | **LongAssignment** |
| **Code:** |  |
| **LOC:** | **150** |
| **Slot(s):** | **3** |

**Title**

Text Message Censoring.

**Background Context**

Email spam, also known as junk email or unsolicited bulk email (UBE), is a subset of electronic spam involving nearly identical messages sent to numerous recipients by email. The messages may contain disguised links that appear to be for familiar websites but in fact lead to phishing web sites or sites that are hosting malware. Spam email may also include malware as scripts or other executable file attachments. Definitions of spam usually include the aspects that email is unsolicited and sent in bulk.

**Program Specifications**

Many people, after a late night, for whatever reason, tend to send emails or text messages that they shouldn’t. Google has experimented with a feature that “blocks” these emails from being sent immediately. (What they do is ask the user some simple math questions before the send operation can execute. Their philosophy is that if someone can’t answer some simple math questions, then maybe the messages they are sending out aren’t the ones they would typically want to send out, under normal conditions.) You don’t want to get in trouble for stealing Google’s idea, but you’ve noticed that the same principle can be applied to text messages. In general, here is what you have noticed:

All text messages sent in between 7:00am and 12:59am (the next day) are reasonably sound messages. But, some messages in between 1:00am and 6:59am tend to be suspect. To reduce your workload, (since censoring requires reading through the whole message), you will ONLY censor messages sent in between 1:00am and 6:59am. All other messages are automatically sent without inspection.

When you inspect the messages sent in between 1:00am and 6:59am, you will censor (block) any message with the following characteristics:

1) The strings “I”, “love”, and “you” appear consecutively, in that order, in any capitalization

2) Three or more misspelled words (you will be given a dictionary of valid “text” words)

3) Has a forbidden word (you will be given a list of these as well)

All comparisons should be done case-insensitive.

***Function details:***

* The email content input from file ***textmsg.txt***. The first line of the input file will have a single positive integer, n (n ≤ 30000), representing the number of words in the dictionary. The next n lines will contain one word from the dictionary each, all in lowercase letters, listed in alphabetical order. (No word will be longer than 29 letters.)
* The next line in the input file will have a single positive integer, m (m ≤ 100), representing the number of forbidden words. These will NOT be in the dictionary. The next m lines will have each of the forbidden words listed, all in lowercase, one word per line, in alphabetical order.
* The following line of the input file will have a single positive integer, t, representing the number of text messages to examine. The rest of the file will contain 2t lines. The relevant information for each text message will be contained in 2 lines. The first line for each text message will contain a time listed in the following format:

*hh:mm AM (or)*

*hh:mm PM*

Note: two digits will be used to represent the hour, only when necessary, otherwise 1 digit will be used. There will always be a space after the time and either AM or PM, and the latter will always be capitalized as shown. This time represents when the user wants to send the text message.

* The next line will contain a positive integer, w (w < 50), representing the number of words in the message, followed by the text message itself and is guaranteed just to contain strings with letters only separated by space.
* For each message, output a header as follows:

*Message #x:*

where x (1 ≤ x ≤ t) is number of the text message. (Note: Always leave a space after the colon.)

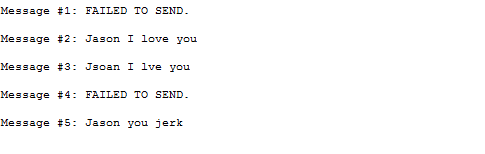
If the message should be censored, then finish the line with the following string:

*FAILED TO SEND.*

Otherwise, simply put the original text of the message (in its original capitalization) to follow.

Separate the output for each case with a blank line.

***Expectation of User interface:***



**Guidelines**