Tokenization (40%)

Problem ID: mt2p2

Write a program that prompts the user for a name of a file containing a sequence of words in each line. The program tokenizes the text, i.e. splits the sequences into tokens using white space as a delimiter. Furthermore, if a comma, period, exclamation mark, or question mark (, . ! ?) is attached to a word, then these punctuation characters are considered separate tokens. You can assume that these are the only punctuation marks and if they appear in the input file, they are always attached to the end of a word (no white space before a punctuation mark).

You are **not** allowed to use any import statement in your solution, except `import typing`.

First example of an input file, `data1.txt`:

```
One Two Three
Four Five Six
Seven Eight Nine
```

Second example of an input file, `data2.txt`:

```
One Two Three? Four
Five, Six Seven Eight!
Nine Ten.
```

Third example of an input file, `data3.txt`:

```
This is a text,
with some
punctuation characters! attached
to some words.
Right?
```

Input

The input is a name of a file to be tokenized, e.g. `data1.txt` or any other name ending in `.txt`. The input file contains 1 to 10 lines, where each line contains a sequence of 1 to 15 words, and the length of each word is 1 to 10 characters. Here, a word is defined as a sequence of upper case and/or lowercase letters, possibly with a comma, period, exclamation mark, or question mark attached to its end.

Note: Your program is not supposed to validate this input, or refuse other input. This is just for your information about the input in the test cases. You do not need to expect input that does not meet these restrictions.

Output

If the input file cannot be opened, no output is generated. Otherwise, the output consists of the following:

1. The word count in a separate line, followed by each word from the input file in a separate line.
2. The token count in a separate line, followed by each token from the input file in a separate line.

Sample Input 1

```
data1.txt
```

Sample Output 1

```
9
One
Two
Three
Four
Five
Six
Seven
Eight
Nine
9
One
Two
Three
Four
Five
Six
Seven
Eight
Nine
```

Sample Input 2

```
data2.txt
```

Sample Output 2

```
10
One
Two
Three?
Four
Five,
Six
Seven
Eight!
Nine
Ten.
14
One
Two
Three
?
Four
Five
,
Six
Seven
Eight
!
Nine
Ten
.
```

Sample Input 3

```
data3.txt
```

Sample Output 3

```
13
This
is
a
text,
with
some
punctuation
characters!
attached
to
some
words.
Right?
17
This
is
a
text
,
with
some
punctuation
characters
!
attached
to
some
words
.
Right
?
```