Deterministic Url and HashTag Segmentation

Deterministic Url and HashTag Segmentation

This problem will introduce you to the segmentation of Domain Names and Social Media HashTags, into English Language words. To give you a quick idea of what segmentation means, here are a few examples of Domain names and Hash Tags which have been segmented.

Domain Name Examples:-

www.checkdomain.com => [check domain]

www.bigrock.com => [big rock]

www.namecheap.com => [name cheap]

www.appledomains.in => [apple domains]

Twitter Hash Tag Examples:

#honestyhour => [honesty hour]

#beinghuman => [being human]

#followback => [follow back]

#socialmedia => [social media]

#30secondstoearth => [30 seconds to earth]

The segmentation should be based on the list of 5000 most common words from here Apart from the words in this list, you should also pick up numbers (both integer and decimal) like 100, 200.10 etc.

At this stage, we are going to use a very simple algorithm for the process. In case the input is a domain name, ignore the www. and/or the extensions (.com,.edu,.org,.in, etc.) In case the input is a hashtag, ignore the first # symbol. Split the input string, into a sequence of tokens. A token can either be:

- A word in from the provided lexicon/dictionary.
- An integer or decimal number.

There might be cases where it might be possible to parse (or split) an input string into tokens in multiple possible ways.

```
currentratesoughttogodown
```

This can be split into:

- current rate sought to go down
- current rates ought to go down.

  thisisinsane

This can be split into:

- this is in sane

- this is insane

Write your splitter in such a way, that as you tokenise a string from left to right; in case there are multiple possible ways to split the string,

select the longest possible string from the left side, such that the remaining string can be split into valid tokens. So, for the two cases above, the appropriate ways to split the strings are:

- current rates ought to go down
- this is insane

In case there is no valid way to split the string into a valid sequence of tokens, output the original string itself, after scrubbing out the # for hashtags, the 'www' and extensions for domain names.

Input Format

First line will contain the number of test cases N

This will be followed by N inputs on separate lines, which will contain twitter hash-tags and domain names, which you need to segment

There will be a file named "words.txt" in the run directory of your program that contains all of the words each seperated by a new line. It is the same file that is linked earlier in the problem statement.

Output Format

(Everything should be in lower case)

```
Segmentation for Input 1
Segmentation for Input 2
          .
          .
Segmentation for Input N
```

Sample Input

```
3
#isittime
www.whatismyname.com
#letusgo
```

Sample Output

```
is it time
what is my name
let us go
```

The sample input is just to get an idea of what to do. Your program is not expected to be able to run it. You can also read the corpus of words by making your program read the file "words.txt" from its current directory.

Please note, that the "words.txt" file, is a list of several common words, but it will not necessarily produce the ideal natural language segmentation for each of the examples, samples or tests. That is the expected behavior: we are only trying to gauge how well you can segment these hashtags and domain names, with this limited list of words.

Scoring

All test cases have equal weightage.

Score = M * (C)/N Where M is the Maximum Score for the test case.
C = Number of correct answers in your output.

N = Total number of tests.