

B. Text Messaging

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Fangy the little walrus, as all the modern walruses, loves to communicate via text messaging. One day he faced the following problem: When he sends large texts, they are split into parts each containing n characters (which is the size of one text message). Thus, whole sentences and words get split!

Fangy did not like it, so he faced the task of breaking the text into minimal messages on his own so that no sentence were broken into pieces when it is sent and the number of text messages to be sent would be minimal. If two consecutive sentences are in different messages, the space between them can be ignored (Fangy does not write this space).

The little walrus's text looks in the following manner:

```
TEXT ::= SENTENCE | SENTENCE SPACE TEXT
SENTENCE ::= WORD SPACE SENTENCE | WORD END
END ::= {'.', '?', '!'}
WORD ::= LETTER | LETTER WORD
LETTER ::= {'a'..'z', 'A'..'Z'}
SPACE ::= ' '
```

SPACE stands for the symbol of a space.

So, how many messages did Fangy send?

Input

The first line contains an integer n , which is the size of one message ($2 \leq n \leq 255$). The second line contains the text. The length of the text does not exceed 10^4 characters. It is guaranteed that the text satisfies the above described format. Specifically, this implies that the text is not empty.

Output

On the first and only line print the number of text messages Fangy will need. If it is impossible to split the text, print "Impossible" without the quotes.

Examples

input
25 Hello. I am a little walrus.
output
2
input
2 How are you?
output
Impossible
input
19 Hello! Do you like fish? Why?

output
3

Note

Let's take a look at the third sample. The text will be split into three messages: "Hello!", "Do you like fish?" and "Why?".