Problem M
TL;DR
Time limit: 2 seconds

Have you ever gotten any of those e-mails that's way too long when you consider what the actual point of the e-mail was? A 1 000 character e-mail could have delivered the same information in 100 characters. Often, the middle of these e-mails is "filler", the sender is trying to butter you up to do something for them. For this problem, write a program that condenses such a block of text down to some number of characters by removing characters from the middle of the text, and tells you how many characters were omitted due to "wordiness" by the author.

For example, the very first paragraph of this problem contains such filler. Suppose we want to condense that 555-character paragraph down to 350 characters. This can be accomplished as follows:

```
Have you ever gotten any of those e-mails that's way too long when you
consider what the actual point of the e-mail was? A 1 000 character
e-mail could have delivered th<218 omitted>t down to some number of
characters by removing characters from the middle of the text, and tells
you how many characters were omitted due to "wordiness" by the author.
```

Here, the `<218 omitted>` characters

```
"e same information in 100 characters. Often, the middle of these
e-mails is "filler", the sender is trying to butter you up to do
something for them. For this problem, write a program that condenses
such a block of tex"
```

added nothing to the problem statement, and they were omitted. The entire first paragraph went from 555 characters down to 350 characters: 169 characters before the `<218 omitted>`, 13 characters for `<218 omitted>`, and 168 characters after `<218 omitted>`.

Your task is the following: given a string s and a value n you must try to reduce s to a string of length n by removing characters from the middle of s and replacing them with the phrase `<# omitted>` (where # is the number of removed characters), or leaving s unchanged if its length is already $\leq n$. The resulting (possibly) reduced string must contain as many characters from the original s as possible. If characters were removed from the middle of s, the number of characters before the "`<# omitted>`" must always be the same as, or one more than, the number of characters after it.

Input

Input begins with a line containing an integer n ($1 \leq n \leq 10\,000$), which is the maximum number of characters allowed in the output. Following this is a single line containing a non-empty string of ASCII printable characters s. The length of s is $\leq 10\,000$. The string s will not start or end with a space character.

Output

If the length of s is $<= n$, output s; otherwise, if s can be modified to be n characters long as described above, output the modified s; otherwise output `impossible`.

Sample Input 1	Sample Output 1
17 ABCDEFGHIJKLMNOPQRSTUVWXYZ	ABC<21 omitted>YZ

Sample Input 2	Sample Output 2
50 Now is the time!	Now is the time!

Sample Input 3	Sample Output 3
25 ABCDEFGHIJKLMNOPQRSTUVWXYZ	ABCDEFG<13 omitted>UVWXYZ

Sample Input 4	Sample Output 4
12 November 8, 2026	impossible