C++ Advanced - Exam (3 July 2021)

For this exam, the code for each task should be a single C++ file, the contents of which you copy-paste into the Judge system.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

You can use C++03 and C++11 features in your code.

1. Format Lines

You are given several **lines of text**, and a **maximum width** value. Your task is to reformat them – merge short lines and break long lines. The lines of the resulting text must:

1. Contain the original words (i.e. words can't be broken apart), in their original order
2. Have at least one word in each of them
3. Not exceed the **maximum width** value (unless this interferes with 1 and 2, i.e. if a word is longer than the maximum width, for that word there must be a line containing only it, even though that line will be wider than the maximum width) – the width of a line is the number of characters on it, including spaces.
4. Be as close to the maximum width as possible (i.e. the number of lines must be the least possible)
5. Have exactly one `' '` (space) character between each two words on a line

A word in the text we will consider any sequence of non-whitespace characters separated by space characters. That is, words consist of combinations of the letters **a-z**, **A-Z**, the digits **0-9**, and the symbols `` `~!@$%^&*()_+{}:"|<>?[];'\,./`` and are separated by one or more `' '` (space) characters and/or new lines. That means that in the text `"hello there, what's up?"`, `"hello"` `"there"` `"what's,"` and `"up?"` are words.

*Simply put, you have some text, remove unnecessary spaces and insert "end of line" when no more words can be added to the line without it becoming longer than the **maximum width** value.*

Input

Lines of the text, until a line containing the string `"###"` (three number signs, aka "hashtags") is entered – this last is not considered part of the text, just an indication that the program should stop reading the text.

On the next line, a single integer will be entered – the maximum width value.

Output

The lines of the formatted text, as described above.

Restrictions

The text will contain no more than **1000** characters (including spaces and `"end of line"` characters).

The **maximum width** will be less than or equal to **100**.

The total running time of your program should be no more than **0.1s**

The total memory allowed for use by your program is **16MB**

Examples

Input	Output
D/1770 L1, popularly known as Lexell's Comet after its orbit computer Anders Johan Lexell, was a comet discovered by astronomer Charles Messier in June 1770.[note 1] It is notable for having passed closer to Earth than any other comet in recorded history, approaching to a distance of only 0.015 astronomical units (2,200,000 km; 1,400,000 mi).[1][3][4] The comet has not been seen since 1770 and is considered a lost comet. ### 38	D/1770 L1, popularly known as Lexell's Comet after its orbit computer Anders Johan Lexell, was a comet discovered by astronomer Charles Messier in June 1770.[note 1] It is notable for having passed closer to Earth than any other comet in recorded history, approaching to a distance of only 0.015 astronomical units (2,200,000 km; 1,400,000 mi).[1][3][4] The comet has not been seen since 1770 and is considered a lost comet.

Input	Output
every word in this text is too wide to fit in the line width ### 1	every word in this text is too wide to fit in the line width