

## B. Wizards and Minimal Spell

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Let's dive into one of the most interesting areas of magic — writing spells. Learning this exciting but challenging science is very troublesome, so now you will not learn the magic words, but only get to know the basic rules of writing spells.

Each spell consists of several lines. The line, whose first non-space character is character "#" is an *amplifying line* and it is responsible for spell power. The remaining lines are common, and determine the effect of the spell.

You came across the text of some spell. Spell was too long, so you cannot understand its meaning. So you want to make it as short as possible without changing the meaning.

The only way to shorten a spell that you know is the removal of some spaces and line breaks. We know that when it comes to texts of spells, the spaces carry meaning only in the amplifying lines, so we should remove all spaces in other lines. Newlines also do not matter, unless any of the two separated lines is amplifying. Thus, if two consecutive lines are not amplifying, they need to be joined into one (i.e. we should concatenate the second line to the first one). Removing spaces in amplifying lines and concatenating the amplifying lines to anything is forbidden.

Note that empty lines must be processed just like all the others: they must be joined to the adjacent non-amplifying lines, or preserved in the output, if they are surrounded with amplifying lines on both sides (i.e. the line above it, if there is one, is amplifying, and the line below it, if there is one, is amplifying too).

For now those are the only instructions for removing unnecessary characters that you have to follow (oh yes, a newline is a character, too).

The input contains the text of the spell, which should be reduced. Remove the extra characters and print the result to the output.

### Input

The input contains multiple lines. All characters in the lines have codes from 32 to 127 (inclusive). Please note that the lines may begin with or end with one or more spaces. The size of the input does not exceed 1048576 ( $= 2^{20}$ ) bytes. Newlines are included in this size.

In the Windows operating system used on the testing computer, a newline is a sequence of characters with codes #13#10. It is guaranteed that after each line of input there is a newline. In particular, the input ends with a newline. Note that the newline is the end of the line, and not the beginning of the next one.

It is guaranteed that the input contains at least one character other than a newline.

It is recommended to organize the input-output line by line, in this case the newlines will be processed correctly by the language means.

### Output

Print the text of the spell where all extra characters are deleted. Please note that each output line should be followed by a newline.

Please be careful: your answers will be validated by comparing them to the jury's answer byte-by-byte. So, all spaces and newlines matter.

### Examples

input

```
#    include <stdio>

using namespace std;
```

```
int main    (    ){
puts("Hello # World"); #
#
}
```

## output

```
#    include <stdio>
usingnamespacestd;intmain(){puts("Hello#World");#
#
}
```

## input

#

#

## output

#

#

## Note

In the first sample the amplifying lines are lines 1 and 7. So, lines 2 to 6 are concatenated to each other, all spaces are deleted from them.

In the second sample the amplifying lines are lines 1 and 3. So, no lines are concatenated to each other.