Problem 3 – Cubic's Messages

Cubic is a veteran soldier from The Great Cubic Army. He has even participated in the Spherical Invasion as a Sergeant First Class. As a veteran, Cubic has some personal security issues – he communicates only trough text messages and sends them in a specific encrypted way, which you must decrypt in order to understand what he is saying.

You will begin receiving lines of input, which will consist of random ASCII characters – Cubic's encrypted lines. After each line you will receive a number – the length of the message he sent. Cubic might send false messages, in an act to confuse his "enemies". You must capture only the messages that follow a certain format.

According to that format the **valid** messages:

- Consist of **m** characters, where **m** is the integer entered after each encrypted line.
- Has only digits before itself in the encrypted line
- Consists only of English alphabet letters
- Has no English alphabet letters after itself in the encrypted line

Any message that **does not follow** the, specified above, rules, is **invalid**, and you must **ignore it**.

After you find **all valid** messages, you need to find their **verification code**. Every message has its own verification code, which Cubic gives in order to verify the message. **Take all the digits before the message** and all the digits **after the message** and consider them as **indexes**. If they are **valid existing** indexes **in the message**, **form a string** with those indexes **taking characters from the message**. If an index is **nonexistent**, put a **space** there. The string you form up is the verification code for the current message.

Input

- The input will always come in the form of 2 lines, except when it is the line terminating the input sequence.
- The first input line will contain random ASCII characters, and the second – a number.
- When the line "**Over!**" is entered, the input sequence ends.

Output

- The output is simple. You must print all the valid messages you've found, each on a new line, and their verification codes, if they have such.
- The format of output is "**{message} == {verificationCode}**".

Constraints

- The input lines can consist of **ANY ASCII** character.
- There will be **NO** such cases as an encrypted message without a number before it.
- The number will be a valid integer in the range [0, 100].
- Allowed time/memory: 100ms/16MB

Examples

Input	Output

`1234test4321`
`4`
`0000oooo0000`
`4`
`Over!`
	`test == est  tse`
	`oooo == oooooooo`

Input	Output

`1wat!`
`3`
`#23asd33`
`3`
`333asd3a`
`3`
`100dun2`
`3`
`Over!`
	`wat == a`
	`dun == uddn`