

# Pattern Matching

A regular expression is a string which contains some normal characters and some meta characters. The meta characters include:

- `.` means any character
- `[c1 - c2]` means any character between c1 and c2 (c1 and c2 are two characters).
- `[^c1 - c2]` means any character not between c1 and c2 (c1 and c2 are two characters).
- `*` means the character before it can occur any times.
- `+` means the character before it can occur any times but at least one times.
- `\` means any character follow should be treated as normal character.

You are to write a program to find the leftmost substring of a given string, so that the substring can match a given regular expression. If there are many substrings of the given string can match the regular expression, and the left positions of these substrings are same, we prefer the longest one.

## Input

Every two lines of the input is a pattern-matching problem. The first line is a regular expression, and the second line is the string to be matched. Any line will be no more than 80 character. A line with only an end will terminate the input.

## Output

For each matching problem, you should give an answer in one line. This line contains the string to be matched, but the leftmost substring that can match the regular expression should be bracketed. If no substring matches the regular expression, print the input string.

## Example

### Input:

```
. *
asdf
f.*d.
sefdfsde
[0-9]+
asd345dsf
[^\*-\*]
**asdf**fasd
b[a-z]*r[s-u]*
abcdefghijklmnopqrstuvwxyz
[T-F]
dfkgjf
end
```

### Output:

```
(asdf)
se(fdfsde)
asd(345)dsf
**(a)sdf**fasd
a(abcdefghijklmnopqrstuvwxyz)
```

dfkgjf