

Czech ACM Student Chapter

Charles University in Prague
Masaryk University
Pavol Jozef Šafárik University in Košice
Slovak University of Technology

Czech Technical University in Prague

Technical University of Ostrava
University of Žilina
Comenius University
University of West Bohemia



CTU Open Contest 2024

Reptile Eggs

`eggs.(c|cpp|py)`, `Eggs.(java|kt)`

The farm generates additional income by producing decorated turkey eggs. The farmers, being excessively creative, design various patterns and apply them to the surface of the eggs. The customer orders are handled with the help of specially trained reptile specimen, often it is a Komodo dragon or a member of another suitable reptile species. The reptile had been trained to understand and execute commands in a specific formal language. A customer order is translated into a command for the reptile which then selects a corresponding group of eggs from the production line. The selected eggs are then packed and prepared for shipment.

The command is a regular expression, and the reptile selects such sequence of eggs on the line that exactly matches the given regular expression. The selected sequence does not have to be contiguous, and it does not have to be specified uniquely by the expression. However, the selected sequence always has to be as long as possible, to keep customers satisfied. The reptile recognizes the beginning and the end of the production line, thus the first egg in the selected sequence is not further from the end of the production line than the last egg in the sequence.

Sometimes, the customer's order, and consequently the regular expression, is so demanding that no unempty sequence of eggs can match the expression. In such situation, the reptile is still able to distinguish between two possible scenarios. In the first scenario, an empty sequence containing no eggs can still match the expression. In the second scenario, even an empty sequence of eggs cannot match the expression. In the first scenario, the reptile remains calm, in the second scenario, the reptile gets dangerously annoyed and has to be calmed down.

Check the reptile's operation and, for each given regular expression, determine the maximum possible number of selected eggs in the shipment.

Below is the specification of the regular expressions that control the reptile's activity:

- Each expression is a finite sequence of lowercase letters of the English alphabet, extended with the characters `?`, `*`, `[`, `]`, `(`, `)`.
- Each letter of the English alphabet represents an egg with a specific pattern corresponding to that letter.
- The question mark (`?`) represents one egg with any pattern.
- The brackets `[` and `]` are always correctly paired, and inside them, there may be only lowercase letters of the English alphabet, at least one. This pair of brackets and the letters within them always represent a single egg on the line, one whose pattern matches any letter inside the brackets.
- The asterisk (`*`) replaces any number of repetitions of the sequence immediately preceding the asterisk. This sequence is either a single letter or a question mark or a portion of the expression enclosed in parentheses `()` or square brackets `[]`. The number of repetitions may also be zero.

- Parentheses `(` and `)` are always correctly paired and are always immediately followed by an asterisk. Inside these parentheses, there is always a regular expression adhering to the rules described above, but it does not contain any parentheses.

Input Specification

The first input line contains integer N ($1 \leq N \leq 1000$). The second input line contains a string of N lowercase English letters representing the types of patterns on the eggs on the production line. Each letter corresponds to one egg. The beginning of the production line corresponds to the beginning of the string. The third input line contains integer M ($1 \leq M \leq 1000$). The fourth input line contains a string of M characters representing the regular expression given to the reptile.

Output Specification

Output single integer E, the maximum possible number of eggs in the shipment selected from the given production line, according to the given regular expression. If the regular expression cannot be matched output -1.

Sample Input 1

```
3
aba
2
a*
```

Output for Sample Input 1

```
2
```

Sample Input 2

```
3
ctu
4
open
```

Output for Sample Input 2

```
-1
```

Sample Input 3

```
4
food
13
([ba]g??t*e)*
```

Output for Sample Input 3

```
0
```

Sample Input 4

```
20
alotofeggcellenteggs
14
[only]*(egg)*?
```

Output for Sample Input 4

```
10
```