

[A. Testrun](#)[B. kolakoski](#)**C. necklace**[D. rocks](#)[E. shipping](#)[Contest Analysis](#)[Questions asked](#) **2**

Submissions

Testrun

0pt Not attempted
0/6 users correct
(0%)

kolakoski

8pt Not attempted
5/7 users correct
(71%)17pt Not attempted
2/5 users correct
(40%)

necklace

16pt Not attempted
10/10 users correct
(100%)29pt Not attempted
9/10 users correct
(90%)

rocks

7pt Not attempted
2/2 users correct
(100%)53pt Not attempted
0/1 users correct
(0%)

shipping

26pt Not attempted
2/6 users correct
(33%)44pt Not attempted
0/1 users correct
(0%)

Top Scores

bmerry	103
Marcin.Smulewicz	71
shik	70
MiSawa	60
ZbanIlya	53
WJMZBMR	45
simonlindholm	45
mk.al13n	45
wan92hy	45
dreamoon	24

Problem C. necklace

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

small
16 points
2 minute timeout

The contest is finished.

large
29 points
10 minute timeout

The contest is finished.

Problem

You've come up with the coolest idea ever for a new fashion trend: customizable necklaces made out of strings with beads that display letters and other characters! The beads appear only on the front of the necklace and read only in one direction, so the string of characters is not circular and irreversible. By itself, this is not really a new idea. The awesome new feature you have in mind is to add a button that lights up some of the beads so that they display a secret message consisting of characters that form a subsequence of the main string of characters. This will have so many applications... just think of the possibilities! And it's so shiny! People are going to love it! Everyone will want their own!

So you announce this product, allowing people to place orders for necklaces by specifying the string of characters to be displayed on the necklace as well as the secret message to be lit up when they press the button. The orders come pouring in! Your idea is even more popular than you expected! How exciting!

Unfortunately, after examining a few orders, you realize that you forgot to check the crucial constraint that the secret message has to be a subsequence of the main necklace string. Without that, the secret message can't always be lit up entirely.

You don't want to disappoint your customers by just telling them that it is impossible to light up their secret messages in the chosen necklace strings. So you decide to offer them an alternative message by finding a substring of their secret message that forms a subsequence of their necklace string, in case they would be satisfied with this shorter version. You want to maximize the length of such a substring.

Given a necklace string **N** and a secret message string **M**, find the maximum length of a substring of **M** that is also a subsequence of **N**.

Input

The input library is called "necklace"; see the sample inputs below for examples in your language. It defines four methods:

- GetNecklaceLength(), which returns the length of the necklace string
- GetNecklaceElement(i), which returns the i-th (0-indexed) element of necklace string
- GetMessageLength(), which returns the length of the secret message
- GetMessageElement(i), which returns the i-th (0-indexed) element of the secret message.

A single call of GetNecklaceElement or GetMessageElement will take up to 0.02 microseconds.

Output

Output one integer - the maximum length of a substring of Message that is also a subsequence of Necklace.

Limits

 $0 \leq \text{GetNecklaceElement}(i), \text{GetMessageElement}(i) \leq 10,000$
 $1 \leq \text{GetNecklaceLength}() \leq 10^9$

Each node will have access to 256MB of RAM and a time limit of 5 seconds. Your solution will run on 100 nodes (both for the small and the large input).

Small input

 $1 \leq \text{GetMessageLength}() \leq 100$

Large input

 $1 \leq \text{GetMessageLength}() \leq 3000$

Sample

Input

See below for sample input files.

Output

For sample input 1:
3
For sample input 2:
1
For sample input 3:
4

Sample input libraries:

Sample input for test 1: [necklace.h](#) [CPP] [necklace.java](#) [Java]

Sample input for test 2: [necklace.h](#) [CPP] [necklace.java](#) [Java]

Sample input for test 3: [necklace.h](#) [CPP] [necklace.java](#) [Java]

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform