**Balkan Olympiad in Informatics 2016**

27th of June - 2nd of July 2016

Nicosia, Cyprus

Day-1 tasks

**Hacker**

Language: en-ISC

# Hacker

A hacker has gained access to Miltos's e-mail account once again. Being fed up with all the intrusions, Miltos finally decides to change his password to a significantly stronger one. In order to do that, he has devised certain criteria that measure the complexity of the password (for example, the amount of repetitiveness), as well as some transformations that hopefully make the password more complex. You are given his old password, W, consisting of N lowercase latin letters. Miltos makes a new password, which initially is just a copy of the old one, and applies a number of changes to it.

Miltos uses three kinds of operations and has asked you to give him valuable information by simulating them. He gives you a list of Q operations of the following kinds:

- *1 i j k*: Check if the substring between positions i and j (inclusive) of the new password is equal to the substring beginning at position k of the new password and having length j - i + 1. If yes, print "Y", else print "N".

- *2 i j k*: Replace the substring that is formed between positions i and j (inclusive) of the new password, with the substring starting at position k of the old password (W) and having length j − i + 1.

- *3 i j*: Replace all letters in the substring between i and j (inclusive) of the new password with the next letter, in a circular fashion. That is, replace "a" with "b", "b" with "c", …, and "z" with "a", for all occurrences of each letter in the substring.

Note that all indices start from 1 (i.e., $1 \le i, j, k \le N$). It is guaranteed that all positions i, j, k are valid. Note also that the length of the new password does not change (it is always equal to N).

## Input

In the first line of the input the old password is given. In the second line the number Q is given and in the next Q lines the operations to simulate are given, each on a separate line.

## Output

The output should be the answers to the operations of type 1, each answer ("Y" or "N") in a separate line.

# Example

## Input

```
aaaaxazyzab
8
1 1 3 2
2 1 3 7
1 1 3 2
1 2 4 8
3 2 4
1 2 4 9
2 7 10 1
1 6 8 8
```

## Output

```
Y
N
Y
Y
Y
```

# Subtasks

- **Subtask 1 (6%):** N, Q ≤ 1 000.

- **Subtask 2 (11%):** N, Q ≤ 300 000. There will only be operations of the first type.

- **Subtask 3 (37%):** N, Q ≤ 300 000. There will only be operations of the first two types.

- **Subtask 4 (46%):** N, Q ≤ 300 000.