Periodic Words (`periodicwords`)

A string s is said to be *periodic* if there exists a string t such that s can be obtained by concatenating multiple (at least 2) copies of t. In other words, s is periodic if $s = t + t + \ldots + t$ for some string $t \neq s$, where $+$ is the string concatenation operation.

You are given a string $A = \overline{a_0 a_1 \ldots a_{N-1}}$ of length N and Q queries of the form l_i, r_i. For each query, determine whether the substring $A[l_i \ldots r_i] = \overline{a_{l_i} a_{l_i+1} \ldots a_{r_i}}$ is a periodic string.

> ☞ Among the attachments of this task you may find a template file `periodicwords.*` with a sample incomplete implementation.

Input

The first line contains an integer N. The second line contains a string S of length N. The third line contains an integer Q. The next Q lines contain the values l_i, r_i, describing the queries.

Output

For each query, print `YES` if the required substring is a periodic string or `NO` if it is not.

Constraints

- $1 \leq N, Q \leq 100\,000$.
- $0 \leq l_i \leq r_i \leq N - 1$.
- The string consist of lowercase letters of the English alphabet.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (10 points) $N, Q \leq 100$.

– **Subtask 3** (20 points) $N, Q \leq 1000$.

– **Subtask 4** (20 points) $N, Q \leq 10\,000$.

– **Subtask 5** (50 points) No additional limitations.

Examples

input	output
14 abacbaabcabccc 5 0 13 0 3 6 11 11 13 6 10	NO NO YES YES NO

Explanation

In the **first query**, it is asked if the whole string is periodic, so the answer is `NO`.

In the **third query**, the substring *abcabc* is periodic, as it can be obtained by concatenating *abc* twice.