Problem A. Prefixuffix

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 megabytes

We consider strings consisting of lowercase letters of the English alphabet in this problem. An initial fragment of a given string is called its *prefix*. A final (terminal) fragment of a given string is called its *suffix*. In particular, the empty string is both a prefix and a suffix of any string. Two strings are *cyclically equivalent* if one of them can be obtained from another by moving its certain suffix from the end of the string to its beginning. For example, the strings *ababba* and *abbaab* are cyclically equivalent, whereas the strings *ababba* and *ababab* are not. In particular, every string is cyclically equivalent to itself.

A string t consisting of n letters is given. We are looking for its prefix p and suffix s of equal length such that:

- p and s are cyclically equivalent,
- the common length of p and s does not exceed $\frac{n}{2}$ (i.e., the prefix p and the suffix s do not overlap in t), and
- the common length of p and s is maximized.

Input

The first line of the standard input contains a single integer n ($1 \le n \le 1\,000\,000$) denoting the length of the string t. The second line of input contains the string t itself, consisting of n lowercase letters of the English alphabet.

Output

Your program should print a single integer in the first and only line of the standard output, namely the common length of the prefix p and the suffix s that we are looking for.

Examples

standard input	standard output
15 ababbababababbaab	6

Problem B. Obscene Words Filter

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

There is a problem to check messages of web-board visitors for the obscene words. Your elder colleagues commit this problem to you. You are to write a program, which check if there is at least one obscene word from the given list in the given text as a substring.

Input

The first line consists of integer n ($1 \le n \le 10000$) — an amount of words. The next n lines contain the list of words that we can't allow to use in our well-educated society. A word may contain only spaces, numbers $[0-9]$ and latin letters $[a-z, A-Z]$. The length of each word doesn't exceed 10000 symbols. The total list of words doesn't exceed $100KB$. Then there is an integer m — the number of lines of the text. A size of the text doesn't exceed $900KB$.

Output

Output the number of line and the number of position separated with a space, where an obscene word occurs for the first time. If there are no obscene words, output"Passed".

Examples

standard input	standard output
5 dear sweetie angel dream baby 8 Had I the heavens embroidered cloths Enwrought with golden and silver light The blue and the dim and the dark cloths Of night and light and the half light I would spread the cloths under your feet But I being poor have only my dreams I have spread my dreams under your feet Tread softly because you tread on my dream	6 31

Problem C. Cipher Message 2

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Muller had been suspecting for a long time that Stierlitz had been sending cipher messages to the USSR from time to time. And now Muller almost got the proof of that. Rummaging through Stierlitz's papers, he found a strange sequence of digits written on a clean sheet of paper. He guessed that it was a cipher message and called Stierlitz for questioning. But Stierlitz calmly answered that the digits were the number of a lottery-ticket that he had written in order not to forget it. Stierlitz had never been so close to a failure: there were the coordinates of Hitler's bunker on the sheet.

For transmitting the data to the center, Stierlitz used the following algorithm:

- The input is a string $s = s_1 s_2 \dots s_n$.
- A key k is chosen; it is a positive integer smaller than n.
- For every symbol s_i of the string, the following procedure is applied:
  1. The string q_i is considered consisting of k onsecutive symbols of the string s starting from the ith: $q_i = s_i s_{i+1} \dots s_{i+k-1}$. If there are less than k symbols till the end of the string, then the remaining symbols are taken from the beginning of the string: $q_i = s_i \dots s_n s_1 \dots s_{i+k-1-n}$.
  2. For the string q_i, the number of its different nonempty substrings m_i is calculated.
- The sequence $m_1, m_2, \dots, m_n$ is the output of the algorithm.

It is not easy to cipher with this algorithm, and how to decode the messages only the Soviet intelligence service knows. You are given a chance to feel yourself the famous Stierlitz for several minutes.

Input

In the first line you are given the key k, $1 \leqslant k \leqslant 1\,000$. The second line contains the string s you are supposed to cipher.

The string consists of lowercase English letters, and its length is strictly greater than k and does not exceed 4000.

Output

Output the numbers $m_1, m_2, \dots, m_n$ separated with spaces.

Examples

standard input	standard output
3 abaccc	5 6 5 3 5 6

Problem D. Palindromic factory

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

Consider string g. Let's call it palindromic generator. Set of palindromes $P(g)$, which are generated by this string are defined as follows:

Let the length of string be n. For each i from 1 to n strings $g[1..i]g[1..i]^r$ and $g[1..i]g[1..i-1]^r$ are included in $P(g)$, where a^r means a written backwards.

For example if g = "*olymp*", then $P(g)$ = "*oo*", "*o*", "*ollo*", "*olo*", "*olyylo*", "*olylo*", "*olymmylo*", "*olymylo*", "*olymppmylo*", "*olympmylo*".

For given generator g and string s you are to find the number of occurences of strings from $P(g)$ in s as substring. More precise, you have to find how many there are such pairs (i, j) that $s[i..j] \in P(g)$.

Input

First line of input contains string g. Second line of input contains string s. Both strings are non-empty and their length not exceed 100000 letters.

Output

Output number of such pairs (i, j) that $s[i..j] \in P(g)$.

Examples

standard input	standard output
olymp olleolleolympmyolylomylo	7

Problem E. Cyclic shifts

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Given a string s. Consider all its cyclic shifts. You are to find lexycographically minimum one.

Input

There is the only string s in input ($1 \le |s| \le 100\,000$) consisting of lowercase latin letters.

Output

Output the only string which is lexycographically minimum cyclic shift of s.

Examples

standard input	standard output
cabacaba	abacabac

Problem F. Cyclic suffixes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	0.5 seconds
Memory limit:	128 mebibytes

Given the string s. Consider all of its cyclic shifts sorted lexycographically. You are to find position of the initial string s in this list.

Input

There is the only string s in input ($1 \le |s| \le 1\,000\,000$) consisting of lowercase latin letters.

Output

Output the only integer - position of string s in described list. If there are multiple possible answers output the minimum one.

Examples

standard input	standard output
abracadabra	3
cabcab	5

Note

Cyclic shifts of string "abracadabra" in lexycographical order:

1. aabracadabr
2. abraabracad
3. abracadabra !
4. acadabraabr
5. adabraabrac
6. braabracada
7. bracadabraa
8. cadabraabra
9. dabraabraca
10. raabracadab
11. racadabraab

Cyclic shifts of string "cabcab" in lexycographical order:

1. abcabc
2. abcabc
3. bcabca
4. bcabca
5. cabcab !
6. cabcab