Codeforces Beta Round #46 (Div. 2)

A. Sleuth

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya plays the sleuth with his friends. The rules of the game are as follows: those who play for the first time, that is Vasya is the sleuth, he should investigate a "crime" and find out what is happening. He can ask any questions whatsoever that can be answered with "Yes" or "No". All the rest agree beforehand to answer the questions like that: if the question's last letter is a vowel, they answer "Yes" and if the last letter is a consonant, they answer "No". Of course, the sleuth knows nothing about it and his task is to understand that.

Unfortunately, Vasya is not very smart. After 5 hours of endless stupid questions everybody except Vasya got bored. That's why Vasya's friends ask you to write a program that would give answers instead of them.

The English alphabet vowels are: A, E, I, O, U, Y

The English alphabet consonants are: B, C, D, F, G, H, J, K, L, M, N, P, Q, R, S, T, V, W, X, Z

Input
The single line contains a question represented by a non-empty line consisting of large and small Latin letters, spaces and a question mark. The line length does not exceed 100. It is guaranteed that the question mark occurs exactly once in the line — as the last symbol and that the line contains at least one letter.

Output
Print answer for the question in a single line: `YES` if the answer is "Yes", `NO` if the answer is "No".

Remember that in the reply to the question the last **letter**, not the last character counts. I. e. the spaces and the question mark do not count as letters.

Sample test(s)

input
`Is it a melon?`
output
`NO`

input
`Is it an apple?`
output
`YES`

input
`  Is     it a banana ?`
output
`YES`

input
`Is   it an apple  and a  banana   simultaneouSLY?`
output
`YES`

B. Sum

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya studies positional numeral systems. Unfortunately, he often forgets to write the base of notation in which the expression is written. Once he saw a note in his notebook saying $a + b = ?$, and that the base of the positional notation wasn't written anywhere. Now Vasya has to choose a base p and regard the expression as written in the base p positional notation. Vasya understood that he can get different results with different bases, and some bases are even invalid. For example, expression $78 + 87$ in the base 16 positional notation is equal to FF_{16}, in the base 15 positional notation it is equal to 110_{15}, in the base 10 one — to 165_{10}, in the base 9 one — to 176_9, and in the base 8 or lesser-based positional notations the expression is invalid as all the numbers should be strictly less than the positional notation base. Vasya got interested in what is the length of the longest possible expression value. Help him to find this length.

The length of a number should be understood as the number of numeric characters in it. For example, the length of the longest answer for $78 + 87 = ?$ is 3. It is calculated like that in the base 15 (110_{15}), base 10 (165_{10}), base 9 (176_9) positional notations, for example, and in some other ones.

Input

The first letter contains two space-separated numbers a and b ($1 \le a, b \le 1000$) which represent the given summands.

Output

Print a single number — the length of the longest answer.

Sample test(s)

input
78 87
output
3

input
1 1
output
2

C. Disposition

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya bought the collected works of a well-known Berland poet Petya in n volumes. The volumes are numbered from 1 to n. He thinks that it does not do to arrange the book simply according to their order. Vasya wants to minimize the number of the disposition's *divisors* — the positive integers i such that for at least one j ($1 \le j \le n$) is true both: $j \ mod \ i = 0$ and at the same time $p(j) \ mod \ i = 0$, where $p(j)$ is the number of the tome that stands on the j-th place and mod is the operation of taking the division remainder. Naturally, one volume can occupy exactly one place and in one place can stand exactly one volume.

Help Vasya — find the volume disposition with the minimum number of divisors.

Input

The first line contains number n ($1 \le n \le 100000$) which represents the number of volumes and free places.

Output

Print n numbers — the sought disposition with the minimum divisor number. The j-th number ($1 \le j \le n$) should be equal to $p(j)$ — the number of tome that stands on the j-th place. If there are several solutions, print any of them.

Sample test(s)

input
2

output
2 1

input
3

output
1 3 2

D. Game

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya and Petya have invented a new game. Vasya takes a stripe consisting of $1 \times n$ square and paints the squares black and white. After that Petya can start moves — during a move he may choose any two neighboring squares of one color and repaint these two squares any way he wants, perhaps in different colors. Petya can only repaint the squares in white and black colors. Petya's aim is to repaint the stripe so that no two neighboring squares were of one color. Help Petya, using the given initial coloring, find the minimum number of moves Petya needs to win.

Input

The first line contains number n ($1 \le n \le 1000$) which represents the stripe's length. The second line contains exactly n symbols — the line's initial coloring. `0` corresponds to a white square, `1` corresponds to a black one.

Output

If Petya cannot win with such an initial coloring, print `-1`. Otherwise print the minimum number of moves Petya needs to win.

Sample test(s)

input
6 111010
output
1

input
5 10001
output
1

input
7 1100010
output
2

input
5 00100
output
2

Note

In the first sample Petya can take squares 1 and 2. He repaints square 1 to black and square 2 to white.

In the second sample Petya can take squares 2 and 3. He repaints square 2 to white and square 3 to black.

E. Common ancestor

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The DNA sequence for every living creature in Berland can be represented as a non-empty line consisting of lowercase Latin letters. Berland scientists found out that all the creatures evolve by stages. During one stage exactly one symbol of the DNA line is replaced by exactly two other ones. At that overall there are n permissible substitutions. The substitution $a_i->b_ic_i$ means that any **one** symbol a_i can be replaced with two symbols b_ic_i. Every substitution could happen an unlimited number of times.

They say that two creatures with DNA sequences s_1 and s_2 can have a common ancestor if there exists such a DNA sequence s_3 that throughout evolution it can result in s_1 and s_2, perhaps after a different number of stages. Your task is to find out by the given s_1 and s_2 whether the creatures possessing such DNA sequences can have a common ancestor. If the answer is positive, you have to find the length of the shortest sequence of the common ancestor's DNA.

Input

The first line contains a non-empty DNA sequence s_1, the second line contains a non-empty DNA sequence s_2. The lengths of these lines do not exceed 50, the lines contain only lowercase Latin letters. The third line contains an integer n $(0 \le n \le 50)$ — the number of permissible substitutions. Then follow n lines each of which describes a substitution in the format $a_i->b_ic_i$. The characters a_i, b_i, and c_i are lowercase Latin letters. Lines s_1 and s_2 can coincide, the list of substitutions can contain similar substitutions.

Output

If s_1 and s_2 cannot have a common ancestor, print -1. Otherwise print the length of the shortest sequence s_3, from which s_1 and s_2 could have evolved.

Sample test(s)

input

```
ababa
aba
2
c->ba
c->cc
```

output

```
2
```

input

```
ababa
aba
7
c->ba
c->cc
e->ab
z->ea
b->ba
d->dd
d->ab
```

output

```
1
```

input

```
ababa
aba
1
c->ba
```

output

```
-1
```