A - Keep Them Separated

Problem ID: keepthemsepar
CPU Time limit: 1 second
Memory limit: 1024 MB
Difficulty: 5.6

Source: The 2018 ICPC Vietna National Programming Conte

Team Socket was defeated by Bash — the Pokenom trainer — and Bash's best Pokenom Chikapu for the $10^9 + 7$-th time. Team Socket realized that Bash and Chikapu were simply too strong together. Now team Socket is devising an evil plan to keep Bash and Chikapu separated! Team Socket has built an evil machine, which can instantly build a rectangular wall or instantly remove a rectangular wall.

Given the locations where Team Socket is going to build and remove walls, can you help Team Socket check whether Bash and Chikapu are separated?

You are given Q queries, numbered from 1 to Q. The i-th query can be one of the following 3 types:

- $1\ x_1\ y_1\ x_2\ y_2$: Team Socket builds a rectangular wall, with sides parallel to the axes, and 2 opposite corners at (x_1, y_1) and (x_2, y_2) $(x_1 \neq x_2, y_1 \neq y_2)$.
- $2\ j$: Team Socket removes the rectangular wall built in the j-th query. It is guaranteed that j-th query is of the 1st type, the wall was built before this query (i.e. $j < i$), and the wall was not removed previously.
- $3\ x_1\ y_1\ x_2\ y_2$: Bash is standing at (x_1, y_1), and Chikapu is standing at (x_2, y_2). Please let Team Socket know if there is a path from Bash to Chikapu. Of course, both Bash and Chikapu cannot walk through any walls.

Input

The first line of input contains exactly one integer Q — the number of queries $(1 \leq Q \leq 10^5)$.

Then Q lines follow, the i-th line is one of 3 types:

- $1\ x_1\ y_1\ x_2\ y_2$
- $2\ j$
- $3\ x_1\ y_1\ x_2\ y_2$

All coordinates in the input file are integers from 1 to 5 000, inclusive. It is guaranteed that:

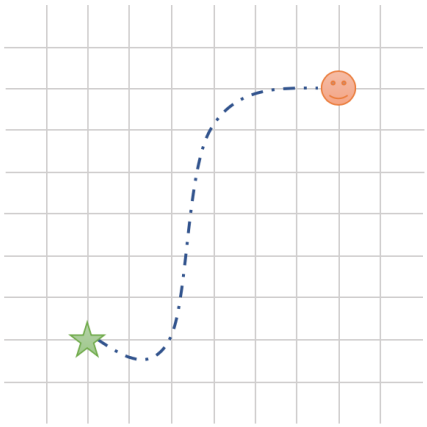
- After each query, no 2 walls have a common point.
- In all queries of 1st type, x_1, y_1, x_2, y_2 are odd numbers.
- In all queries of 3rd type, x_1, y_1, x_2, y_2 are even numbers.

Output

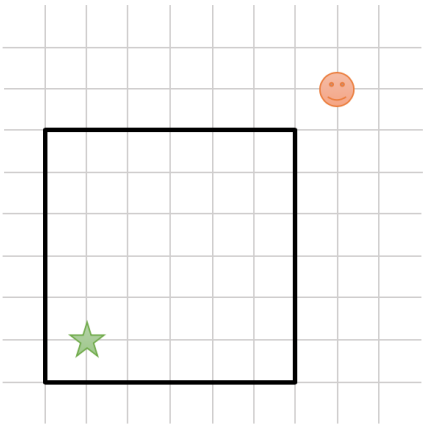
For each query of third type, print a character 'Y' if there is a path from Bash to Chikapu. Otherwise, print a character 'N'. Please note that this problem uses case-sensitive checker.

Sample clarification

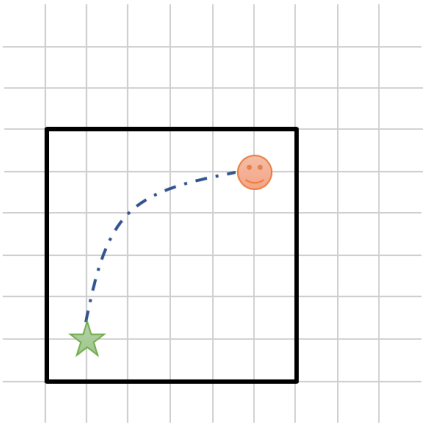
Query 1



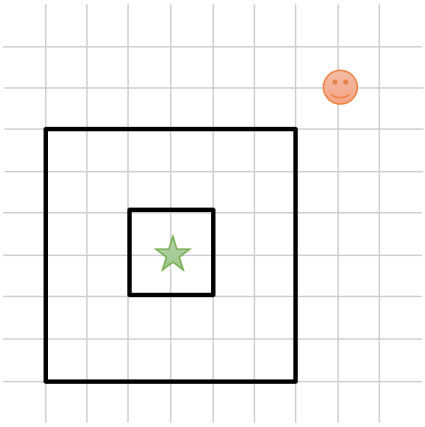
Query 3



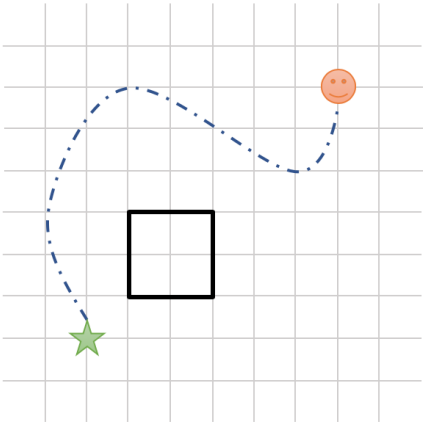
Query 4



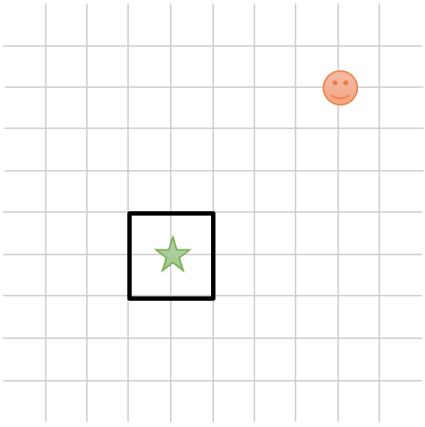
Query 6



Query 8



Query 9



Sample Input 1

```
9
3 2 2 8 8
1 1 1 7 7
3 2 2 8 8
3 2 2 6 6
1 3 3 5 5
3 4 4 8 8
2 2
3 2 2 8 8
3 4 4 8 8
```

Sample Output 1

```
YNYNYN
```

B - The Values You Can Make

CodeForces - 687C

Pari wants to buy an expensive chocolate from Arya. She has n coins, the value of the i-th coin is c_i. The price of the chocolate is k, so Pari will take a subset of her coins with sum equal to k and give it to Arya.

Looking at her coins, a question came to her mind: after giving the coins to Arya, what values does Arya can make with them? She is jealous and she doesn't want Arya to make a lot of values. So she wants to know all the values x, such that Arya will be able to make x using some subset of coins with the sum k.

Formally, Pari wants to know the values x such that there exists a subset of coins with the sum k such that some subset of this subset has the sum x, i.e. there is exists some way to pay for the chocolate, such that Arya will be able to make the sum x using these coins.

Input

The first line contains two integers n and k ($1 \le n, k \le 500$) — the number of coins and the price of the chocolate, respectively.

Next line will contain n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 500$) — the values of Pari's coins.

It's guaranteed that one can make value k using these coins.

Output

First line of the output must contain a single integer q— the numbe
suitable values x. Then print q integers in ascending order — the val
that Arya can make for some subset of coins of Pari that pays for the
chocolate.

Examples

Input
6 18 5 6 1 10 12 2
Output
16 0 1 2 3 5 6 7 8 10 11 12 13 15 16 17 18

Input
3 50 25 25 50
Output
3 0 25 50

C - APIO - ROADS

Z_trening - 1394

The Kingdom of New Asia contains **N** villages connected by **M** roads. Some roads are made of cobblestones, and others are made of concrete. Keeping roads free-of-charge needs a lot of money, and it seems impossible for the Kingdom to maintain every road. A new road maintaining plan is needed.

The King has decided that the Kingdom will keep as few free roads as possible, but every two distinct villages must be connected by one and only one path of free roads. Also, although concrete roads are more suitable for modern traffic, the King thinks walking on cobblestones is interesting. As a result, he decides that exactly **K** cobblestone roads will be kept free.

For example, suppose the villages and the roads in New Asia are as in Figure 1a. If the King wants to keep two cobblestone roads free, then the Kingdom can keep roads (1,2), (2,3), (3,4), and (3,5) free as in Figure 1b. This plan satisfies the King's criteria because (1) every two villages are connected via one and only one path of free roads, (2) there are as few free roads as possible, and (3) there are exactly two cobblestone roads: (2,3) and (3,4).

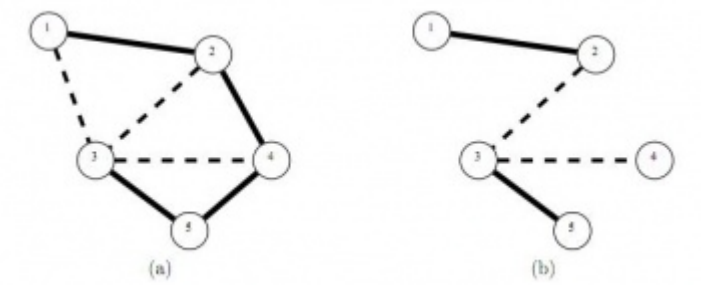


Figure 1: (a) An example configuration of villages and roads in the Kingdom of New Asia. Solid lines denote concrete roads, and dashed lines denote cobblestone roads. (b) A road maintaining plan that ke two cobblestone roads free. Only free roads are shown.

Task

Given a description of roads in New Asia and the number of cobbles roads that the King wants to keep free, write a program to determine if

there is a road maintaining plan that satisfies the King's criteria, and output a valid plan if there is one.

INPUT:
The first line contains three integers separated by one space:
• **N**, the number of villages ($1 \le \mathbf{N} \le 20,000$),
• **M**, the number of roads ($1 \le \mathbf{M} \le 100,000$), and
• **K**, the number of cobblestone roads the King wants to keep free ($0 \le \mathbf{K} \le \mathbf{N} - 1$).
The following **M** lines describes the roads in New Asia, which are numbered 1 to **M**. The (i + 1)st line describes Road i. It contains three intergers separated by one space:
• u_i and v_i, the two villages connected by Road i. Villages are numbered 1 to **N**, and
• c_i, the type of Road i; $c_i = 0$ if Road i is made of cobblestone, and $c_i = 1$ if it is made of concrete.
There will be no more than one road connecting a pair of villages.

OUTPUT:
If there is no road maintaining plan that satisfies the King's criteria, your program should print **no solution** on the first line of the output.

Otherwise, your program should output a valid road maintaining plan by listing roads that will be kept free, one road per line. To list a road, print the line in the input that describes it. The roads can be listed in any order. If there are more than one valid plan, you can output any such plan.

Input:
```
5 7 2
1 3 0
4 5 1
3 2 0
5 3 1
4 3 0
1 2 1
4 2 1
```

Output:
```
3 2 0
4 3 0
1 2 1
5 3 1
```

Scoring
The score for each input scenario will be 100% if the correct answer is outputed and 0% otherwise. In test scenarios worthing 20 points, **K** will be at most 10.

D - Journey CodeForces - 839C

There are n cities and $n - 1$ roads in the Seven Kingdoms, each road connects two cities and we can reach any city from any other by the roads.

Theon and Yara Greyjoy are on a horse in the first city, they are starting traveling through the roads. But the weather is foggy, so they can't see where the horse brings them. When the horse reaches a city (including the first one), it goes to one of the cities connected to the current city. But it is a strange horse, it only goes to cities in which they weren't before. In each such city, the horse goes with equal probabilities and it stops when there are no such cities.

Let the length of each road be 1. The journey starts in the city 1. What is the expected length (expected value of length) of their journey? You can read about expected (average) value by the link https://en.wikipedia.org/wiki/Expected_value.

Input

The first line contains a single integer n ($1 \le n \le 100000$) — number of cities.

Then $n - 1$ lines follow. The i-th line of these lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) — the cities connected by the i-th road.

It is guaranteed that one can reach any city from any other by the roads.

Output

Print a number — the expected length of their journey. The journey starts in the city 1.

Your answer will be considered correct if its absolute or relative error

does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{max(1,b)} \leq 10^{-6}$.

Examples

Input
4 1 2 1 3 2 4
Output
1.500000000000000

Input
5 1 2 1 3 3 4 2 5
Output
2.000000000000000

Note

In the first sample, their journey may end in cities 3 or 4 with equal probability. The distance to city 3 is 1 and to city 4 is 2, so the expected length is 1.5.

In the second sample, their journey may end in city 4 or 5. The dista to the both cities is 2, so the expected length is 2.

E - Alyona and mex CodeForces - 739A

Alyona's mother wants to present an array of n non-negative integers to Alyona. The array should be special.

Alyona is a capricious girl so after she gets the array, she inspects m of its subarrays. Subarray is a set of some subsequent elements of the array. The i-th subarray is described with two integers l_i and r_i, and its elements are $a[l_i], a[l_i + 1], ..., a[r_i]$.

Alyona is going to find mex for each of the chosen subarrays. Among these m mexes the girl is going to find the smallest. She wants this minimum mex to be as large as possible.

You are to find an array a of n elements so that the minimum mex among those chosen by Alyona subarrays is as large as possible.

The mex of a set S is a minimum possible non-negative integer that is not in S.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$).

The next m lines contain information about the subarrays chosen by Alyona. The i-th of these lines contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), that describe the subarray $a[l_i], a[l_i + 1], ..., a[r_i]$.

Output

In the first line print single integer — the maximum possible minin mex.

In the second line print n integers — the array a. All the elements in a should be between 0 and 10^9.

It is guaranteed that there is an optimal answer in which all the elements in a are between 0 and 10^9.

If there are multiple solutions, print any of them.

Examples

Input
5 3 1 3 2 5 4 5
Output
2 1 0 2 1 0

Input
4 2 1 4 2 4
Output
3 5 2 0 1

Note

The first example: the mex of the subarray (1, 3) is equal to 3, the mex of the subarray (2, 5) is equal to 3, the mex of the subarray (4, 5) is equal to 2 as well, thus the minumal mex among the subarrays chosen by Al is equal to 2.

F - Dima and Bacteria CodeForces - 400D

Dima took up the biology of bacteria, as a result of his experiments, he invented k types of bacteria. Overall, there are n bacteria at his laboratory right now, and the number of bacteria of type i equals c_i. For convenience, we will assume that all the bacteria are numbered from 1 to n. The bacteria of type c_i are numbered from $(\sum_{k=1}^{i-1} c_k) + 1$ to $\sum_{k=1}^{i} c_k$.

With the help of special equipment Dima can move energy from some bacteria into some other one. Of course, the use of such equipment is not free. Dima knows m ways to move energy from some bacteria to another one. The way with number i can be described with integers u_i, v_i and x_i mean that this way allows moving energy from bacteria with number u_i to bacteria with number v_i or vice versa for x_i dollars.

Dima's Chef (Inna) calls the type-distribution correct if there is a way (may be non-direct) to move energy from any bacteria of the particular type to any other bacteria of the same type (between any two bacteria of the same type) for zero cost.

As for correct type-distribution the cost of moving the energy depends only on the types of bacteria help Inna to determine is the type-distribution correct? If it is, print the matrix d with size $k \times k$. Cell $d[i][j]$ of this matrix must be equal to the minimal possible cost of energy-moving from bacteria with type i to bacteria with type j.

Input

The first line contains three integers n, m, k ($1 \le n \le 10^5$; $0 \le m \le 10^5$; $1 \le k \le 500$). The next line contains k integers $c_1, c_2, ..., c_k$ ($1 \le c_i \le n$). Each of next m lines contains three integers u_i, v_i, x_i ($1 \le u_i, v_i \le 10^5$; $0 \le x_i \le 10^4$ is guaranteed that $\sum_{i=1}^{k} c_i = n$.

Output

If Dima's type-distribution is correct, print string «Yes», and then k lines: in the i-th line print integers $d[i][1], d[i][2], ..., d[i][k]$ ($d[i][i] = 0$). If there is no way to move energy from bacteria i to bacteria j appropriate $d[i][j]$ must equal to -1. If the type-distribution isn't correct print «No».

Examples

Input
4 4 2 1 3 2 3 0 3 4 0 2 4 1 2 1 2
Output
Yes 0 2 2 0

Input
3 1 2 2 1 1 2 0
Output
Yes 0 -1 -1 0

Input
3 2 2 2 1 1 2 0 2 3 1
Output

```
Yes
0 1
1 0
```

Input
3 0 2 1 2
Output
No

G - Remainders Game CodeForces - 687B

Today Pari and Arya are playing a game called Remainders.

Pari chooses two positive integer x and k, and tells Arya k but not x. Arya have to find the value $x \mod k$. There are n ancient numbers $c_1, c_2, \ldots, c_n$ and Pari has to tell Arya $x \mod c_i$ if Arya wants. Given k and the ancient values, tell us if Arya has a winning strategy independent of value of x or not. Formally, is it true that Arya can understand the value $x \mod k$ for any positive integer x?

Note, that $x \mod y$ means the remainder of x after dividing it by y.

Input

The first line of the input contains two integers n and k ($1 \le n, k \le 1\,000\,000$) — the number of ancient integers and value k that is chosen by Pari.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 1\,000\,000$).

Output

Print "Yes" (without quotes) if Arya has a winning strategy independent of value of x, or "No" (without quotes) otherwise.

Examples

Input
4 5 2 3 5 12
Output

Yes

Input
2 7 2 3
Output
No

Note

In the first sample, Arya can understand $x \mod 5$ because 5 is one of the ancient numbers.

In the second sample, Arya can't be sure what $x \mod 7$ is. For example 1 and 7 have the same remainders after dividing by 2 and 3, but they differ in remainders after dividing by 7.

H - Sonya and Queries CodeForces - 713A

Today Sonya learned about long integers and invited all her friends to share the fun. Sonya has an initially empty multiset with integers. Friends give her t queries, each of one of the following type:

1. + a_i — add non-negative integer a_i to the multiset. Note, that she has a multiset, thus there may be many occurrences of the same integer.
2. - a_i — delete a single occurrence of non-negative integer a_i from the multiset. It's guaranteed, that there is at least one a_i in the multiset.
3. ? s — count the number of integers in the multiset (with repetitions) that match some pattern s consisting of 0 and 1. In the pattern, 0 stands for the even digits, while 1 stands for the odd. Integer x matches the pattern s, if the parity of the i-th from the right digit in decimal notation matches the i-th from the right digit of the pattern. If the pattern is shorter than this integer, it's supplemented with 0-s from the left. Similarly, if the integer is shorter than the pattern its decimal notation is supplemented with the 0-s from the left.

For example, if the pattern is $s = 010$, than integers 92, 2212, 50 and 414 match the pattern, while integers 3, 110, 25 and 1030 do not.

Input

The first line of the input contains an integer t ($1 \le t \le 100\,000$) — the number of operation Sonya has to perform.

Next t lines provide the descriptions of the queries in order they appear in the input file.
The i-th row starts with a character c_i — the type of the corresponding operation. If c_i is equal to '+' or '-' then it's followed by a space and an integer a_i ($0 \le a_i < 10^{18}$) given without leading zeroes (unless it's 0). If c_i equals '?' then it's followed by a space and a sequence of zeroes and onse, giving the pattern of length no more than 18.

It's guaranteed that there will be at least one query of type '?'.

It's guaranteed that any time some integer is removed from the multiset, there will be at least one occurrence of this integer in it.

Output

For each query of the third type print the number of integers matching the given pattern. Each integer is counted as many times, as it appears in the multiset at this moment of time.

Examples

Input
12 + 1 + 241 ? 1 + 361 - 241 ? 0101 + 101 ? 101 - 101 ? 101 + 4000 ? 0
Output
2

```
1
2
1
1
```

Input
4 + 200 + 200 - 200 ? 0
Output
1

Note

Consider the integers matching the patterns from the queries of the third type. Queries are numbered in the order they appear in the input.

1. 1 and 241.
2. 361.
3. 101 and 361.
4. 361.
5. 4000.

I - Cloud of Hashtags CodeForces - 777D

Vasya is an administrator of a public page of organization "Mouse and keyboard" and his everyday duty is to publish news from the world of competitive programming. For each news he also creates a list of hashtags to make searching for a particular topic more comfortable. For the purpose of this problem we define hashtag as a string consisting of lowercase English letters and exactly one symbol '#' located at the beginning of the string. The length of the hashtag is defined as the number of symbols in it without the symbol '#'.

The head administrator of the page told Vasya that hashtags should go in lexicographical order (take a look at the notes section for the definition).

Vasya is lazy so he doesn't want to actually change the order of hashtags in already published news. Instead, he decided to delete some suffixes (consecutive characters at the end of the string) of some of the hashtags. He is allowed to delete any number of characters, even the whole string except for the symbol '#'. Vasya wants to pick such a way to delete suffixes that the total number of deleted symbols is minimum possible. If there are several optimal solutions, he is fine with any of them.

Input

The first line of the input contains a single integer n ($1 \le n \le 500\,000$) — the number of hashtags being edited now.

Each of the next n lines contains exactly one hashtag of positive length.

It is guaranteed that the total length of all hashtags (i.e. the total length of the string except for characters '#') won't exceed 500 000.

Output

Print the resulting hashtags in any of the optimal solutions.

Examples

Input
3 #book #bigtown #big
Output
#b #big #big

Input
3 #book #cool #cold
Output
#book #co #cold

Input
4 #car #cart #art #at
Output
#art #at

Input
3 #apple #apple #fruit
Output
#apple #apple #fruit

Note

Word $a_1, a_2, ..., a_m$ of length m is lexicographically not greater than word $b_1, b_2, ..., b_k$ of length k, if one of two conditions hold:

- at first position i, such that $a_i \neq b_i$, the character a_i goes earlier in the alphabet than character b_i, i.e. a has smaller character than b in the first position where they differ;
- if there is no such position i and $m \leq k$, i.e. the first word is a prefix of the second or two words are equal.

The sequence of words is said to be sorted in lexicographical order if each word (except the last one) is lexicographically not greater than the next word.

For the words consisting of lowercase English letters the lexicographical order coincides with the alphabet word order in the dictionary.

According to the above definition, if a hashtag consisting of one character '#' it is lexicographically not greater than any other valid hashtag. That's why in the third sample we can't keep first two hashtags unchanged and shorten the other two.

J - A Story with Strings CodeChef - SSTORY

Lira and her best friend are now ending highschool and soon they both will head separate ways...

Lira will obviously study Computer Science while her best friend will study Arts... (apparentely, opposites attract, alikes repel right?)

However, as Lira didn't want to lose contact with her, she decided to mix Art and Computer Science one last time and amaze her friend with Computer Science, so she could have a geek coding partner.

To amaze her, she encoded a message in two strings, which she presented to her friend:

Lira: "So, Mary, as you can see, we have two strings here, yes? We have a first longer string, **S1**, which encodes our common path together, and is represented by some letters which represent our lives so far as well all the events we lived together.

The second string, **S2**, will be as long as the first one, or possibly shorter and it is here to represent what the future might bring to both of us :) It is also only composed of letters, which represent some life events we might still experience together in the future.

What I would like you to do, is to find out if the future will be just like now, and, for that,you need to find a contiguous sequence of letters in both of these two strings.

Obviously, as I want our friendship to last very, very much, I ask you to find the longest sequence of life events we might still experience together and tell me which sequence is it and how long is it :D

However, if fate doesn't want us to be together, simply tell me the number 0 and we shall both head separate ways forever.

Turns out, that after Mary solved the problem, she became so intere in CS herself, that both she and Lira are now working together at a v reputable software company. :)

Input

The input will consist of two strings composed only of the letters **a..z**, both of which are at most 250.000 characters in length.

Output

The output should be as stated.

If there is a common life event in Mary and Lira's lives, you should output it on a single line and, on another line, you should output the length of the event.

Otherwise, simply output 0.

On the case where there are multiple answers, the first common event to be found on the second string, should be written to output, followed by its length. See example 3 for details.

Constraints

- **1 ≤ |S1| ≤ 250000**
- **1 ≤ |S2| ≤ 250000**
- **1 ≤ |S2| ≤ |S1|**
- **The last constraint simply means that S2 will never be larger than S1**

Example

Input 1:

```
adsyufsfdsfdsf
fdyusgfdfyu
```

Output 1:

```
fd
2
```

Input 2:

```
adsyufsfdsfdsf
```

```
bbbbbbbbb
```

Output 2:
```
0
```

Input 3:
```
abcdef
defabc
```

Output 3:
```
def
3
```

Explanation

Example case 1. The longest life event is **"fd"** whose length is 2.

Example case 2. The girls will follow separate lives.

Example case 3. On the case where multiple events can happen, **Mary will choose the event which is closer to happen in the future**, and, as such, answer is "def".

Note: The Mary in this problem, is actually a small tribute to my coder friend Mary Kuchumova, which, hopefully will return to Codechef and get back in touch again. :D Mary, you're missed!