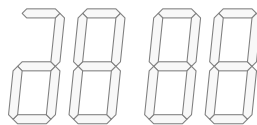


**Wrist watch**

Task ID: watch	Time limit: 0.1 s	Memory limit: 64 MB
----------------	-------------------	---------------------

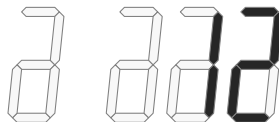
You have recently bought a wrist watch with an LCD display. It displays the time in the 24-hour clock format using four blocks, one per digit.



Three of the blocks consist of seven stripes, each of them is either switched off (has the same color as the background) or on (the stripe becomes black). The picture below shows how different digits are displayed by this kind of block.



The fourth block displays the most significant digit of an hour. If this digit is zero, then the block is entirely switched off, otherwise, the digit is shown in the same manner as above (see picture below).



You want to check if everything works, that is, whether the watch has some broken stripes. A stripe is broken if it is always switched on or always switched off.

You want to start the verification at a certain time. Write a program which determines how much time it should take to fully verify if every stripe works.

Input

The input consists of a single line with the start time of the verification in the format of HH:MM, that is, first the hours are written, then the minutes, both with leading zeros $00 \leq HH \leq 23$, $00 \leq MM \leq 59$. Note that after displaying 23:59, the time 00:00 follows.

Output

Output a single line with a single integer, the number of minutes the verification should take if started at the specified time.

Sample tests

input	output
00:00	1200
input	output
02:39	1041



Strings Again

Task ID: strings	Time limit: 0.15 s	Memory limit: 64 MB
-------------------------	--------------------	---------------------

In this task we will be working with strings of lowercase English letters.

The alphabet in this task will be cyclic, i.e., **z** is followed by **a**. For two letters α and β let $d(\alpha, \beta)$ be their distance in the cyclic alphabet – that is, the smaller of the two possible distances. For example, $d(\mathbf{b}, \mathbf{c}) = d(\mathbf{c}, \mathbf{b}) = 1$, and $d(\mathbf{z}, \mathbf{b}) = 2$.

Let $A = \alpha_1\alpha_2 \dots \alpha_n$ and $B = \beta_1\beta_2 \dots \beta_n$ be two strings of the same length n . Their distance $d(A, B)$ can be computed as the sum of $d(\alpha_i, \beta_i)$ over all i . For example, $d(\mathbf{aba}, \mathbf{aca}) = 1$, $d(\mathbf{aba}, \mathbf{zbz}) = 2$, and $d(\mathbf{abc}, \mathbf{xyz}) = d(\mathbf{a}, \mathbf{x}) + d(\mathbf{b}, \mathbf{y}) + d(\mathbf{c}, \mathbf{z}) = 3 + 3 + 3 = 9$.

The *cyclic shift* of a string $A = \alpha_1\alpha_2 \dots \alpha_n$ by k letters is the string

$$A \hookrightarrow k = \alpha_{n-k+1}\alpha_{n-k+2} \dots \alpha_n \alpha_1 \alpha_2 \dots \alpha_{n-k}$$

Note that in the shifted string each letter is k positions to the right of its original location (wrapping around if necessary).

The *total cyclic distance* t between two strings A and B of equal length is then the sum of the distances of all pairs of possible shifts. In other words, we take each cyclic shift of A , pair it with each cyclic shift of B , and for each such pair of shifted strings we compute their distance. Formally:

$$t = \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} d(A \hookrightarrow k, B \hookrightarrow l)$$

Your task is to calculate the *total cyclic distance* t between two given strings A and B .

Input

The input consists of two lines. The first line contains the string A and the second line contains the string B . Both strings are of the same length n , $1 \leq n \leq 100\,000$ and consist of lowercase English letters only.

For testcases, which are together worth at least 40% of the points, $n \leq 1000$.

Output

Output exactly one line with a single integer t , the *total cyclic distance*.

Sample tests

input	output
a b	1
input	output
ab ac	8

Consider the second example. The cyclic shifts of the string **ab** are **ab** and **ba**.

The cyclic shifts of **ac** are **ac** and **ca**.

The answer then is $T = d(\mathbf{ab}, \mathbf{ac}) + d(\mathbf{ab}, \mathbf{ca}) + d(\mathbf{ba}, \mathbf{ac}) + d(\mathbf{ba}, \mathbf{ca}) = 1 + 3 + 3 + 1 = 8$.



DNA Analysis

Task ID: dna	Time limit: 0.3 s	Memory limit: 64 MB
--------------	-------------------	---------------------

Biologists of Karelia Mutation Project (KMP) have recently started a new project that is intended to prove that humans and mammoths are close relatives. To prove this statement, the biologists are planning to carry out the following experiment.

The human DNA is compared to the mammoth DNA. The DNA is split into pieces of length n , and pieces are consecutively compared. Since some mutations are possible, the comparison is carried out as follows.

Consider a string α . We say that α *mutates* to β if there is a substring y in α which can be reversed to get β out of it. Formally, let $\alpha = xyz$ for some (possibly empty) strings x , y and z , then it can mutate to $\beta = xy^Rz$ where y^R is the string y reversed¹.

We will make an example using $\alpha = \text{"ATTA"}$. If we reverse the last three letters, i.e. using $x = \text{"A"}$, $y = \text{"TTA"}$ and $z = \text{" "}$ (the empty string) we get the mutated string $\beta = \text{"AATT"}$.

Moreover, we say that strings α and β are *similar* if α can be transformed to β by at most 4 **mutations**.

Given two pieces of DNA, check whether they are similar.

Input

The input consists of two lines with one string of length n each, the two DNA sequences. Both strings will only contain the characters 'A', 'D', 'G' and 'T' only. It holds that $1 \leq n \leq 30$.

Output

The output must consist of a single line with either the word "Similar" if the strings are similar, or "Different" if they are not. Note that all quotes are for clarity only, they do not appear in the input and must not appear in the output either.

Partial Scoring

For testcases, which are together worth at least 30% of the points, it holds that $n \leq 7$.

Sample tests

input	output
ATGAATGA AGGAATTA	Similar
input	output
ATGAATGAATGA TTTAAAAAGGG	Different

In the first example two mutations are enough. They can be executed as follows:

$\text{"ATGAATGA"} \rightarrow \text{"AGTAAGTA"} \rightarrow \text{"AGGAATTA"}$.

¹For example $\text{"abc"}^R = \text{"cba"}$.



Important Roads

Task ID: <code>important</code>	Time limit: 0.4 s	Memory limit: 64 MB
---------------------------------	-------------------	---------------------

The city where Georgie lives has n junctions some of which are connected by m bidirectional roads.

Every day Georgie drives from his home to work and back by the fastest possible way. But the roads in the city where Georgie lives are very bad, so they are very often closed for repair. Georgie noticed that when some roads are closed he still can get from home to work in the same time as if all roads were available.

But there are such roads that, if they are closed for repair, the time Georgie needs to get from home to work strictly increases or he might not even get to work at all anymore. Georgie calls such roads *important*.

Help Georgie to find all important roads in the city.

Input

The first line of the input contains two integers n and m , $2 \leq n \leq 20\,000$, $1 \leq m \leq 100\,000$ separated by a space. They describe the number of junctions and roads in the city where Georgie lives, respectively. The junctions are numbered from 1 to n and the roads are numbered from 1 to m . Georgie lives at junction 1 and works at junction n .

The following m lines each contain the description of a road. The i -th line contains three integers a_i, b_i, t_i , $1 \leq a_i \neq b_i \leq n$, $1 \leq t_i \leq 100\,000$, separated by a single space each. They describe the start and end junction of the i -th road and the time Georgie needs to drive along it. There are no two roads connecting the same junctions and no road connects a junction to itself. It is guaranteed that if all roads are available, Georgie can get from home to work.

Output

The first line of the output must contain a single integer l , the number of important roads. The second line of the output must contain l numbers, separated by spaces, the numbers of important roads in increasing order.

Partial Scoring

For testcases, which are together worth at least 30% of the points, it holds that $n, m \leq 500$.

Sample tests

input

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

output

```
2
5 7
```