Codeforces Round #349 (Div. 1)

A. Reberland Linguistics

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

First-rate specialists graduate from Berland State Institute of Peace and Friendship. You are one of the most talented students in this university. The education is not easy because you need to have fundamental knowledge in different areas, which sometimes are not related to each other.

For example, you should know linguistics very well. You learn a structure of Reberland language as foreign language. In this language words are constructed according to the following rules. First you need to choose the "root" of the word — some string which has more than 4 letters. Then several strings with the length 2 or 3 symbols are appended to this word. The only restriction — *it is not allowed to append the same string twice in a row*. All these strings are considered to be suffixes of the word (this time we use word "suffix" to describe a morpheme but not the few last characters of the string as you may used to).

Here is one exercise that you have found in your task list. You are given the word s. Find all distinct strings with the length 2 or 3, which can be suffixes of this word according to the word constructing rules in Reberland language.

Two strings are considered distinct if they have different length or there is a position in which corresponding characters do not match.

Let's look at the example: the word $abacabaca$ is given. This word can be obtained in the following ways: $\overline{abacabaca}, \overline{abacaba}\widehat{ca}, \overline{abacab}\widehat{aca}, \overline{abaca}\widehat{ba}\widehat{ca}$, where the root of the word is overlined, and suffixes are marked by "corners". Thus, the set of possible suffixes for this word is $\{aca, ba, ca\}$.

Input

The only line contains a string s ($5 \le |s| \le 10^4$) consisting of lowercase English letters.

Output

On the first line print integer k — a number of distinct possible suffixes. On the next k lines print suffixes.

Print suffixes in lexicographical (alphabetical) order.

Examples

input

```
abacabaca
```

output

```
3
aca
ba
ca
```

input

```
abaca
```

output

```
0
```

Note

The first test was analysed in the problem statement.

In the second example the length of the string equals 5. The length of the root equals 5, so no string can be used as a suffix.

B. World Tour

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

A famous sculptor Cicasso goes to a world tour!

Well, it is not actually a world-wide. But not everyone should have the opportunity to see works of sculptor, shouldn't he? Otherwise there will be no any exclusivity. So Cicasso will entirely hold the world tour in his native country — Berland.

Cicasso is very devoted to his work and he wants to be distracted as little as possible. Therefore he will visit only four cities. These cities will be different, so no one could think that he has "favourites". Of course, to save money, he will chose the shortest paths between these cities. But as you have probably guessed, Cicasso is a weird person. Although he doesn't like to organize exhibitions, he likes to travel around the country and enjoy its scenery. So he wants the total distance which he will travel to be as large as possible. However, the sculptor is bad in planning, so he asks you for help.

There are n cities and m one-way roads in Berland. You have to choose four different cities, which Cicasso will visit and also determine the order in which he will visit them. So that the total distance he will travel, if he visits cities in your order, starting from the first city in your list, and ending in the last, choosing each time the shortest route between a pair of cities — will be the largest.

Note that intermediate routes may pass through the cities, which are assigned to the tour, as well as pass twice through the same city. For example, the tour can look like that: $\overline{1}, 2, 3, 4, \overline{5}, 4, 3, \overline{2}, 3, \overline{4}$. Four cities in the order of visiting marked as overlines: $[1, 5, 2, 4]$.

Note that Berland is a high-tech country. So using nanotechnologies all roads were altered so that they have the same length. For the same reason moving using regular cars is not very popular in the country, and it can happen that there are such pairs of cities, one of which generally can not be reached by car from the other one. However, Cicasso is very conservative and cannot travel without the car. Choose cities so that the sculptor can make the tour using only the automobile. It is guaranteed that it is always possible to do.

Input

In the first line there is a pair of integers n and m ($4 \le n \le 3000, 3 \le m \le 5000$) — a number of cities and one-way roads in Berland.

Each of the next m lines contains a pair of integers u_i, v_i ($1 \le u_i, v_i \le n$) — a one-way road from the city u_i to the city v_i. Note that u_i and v_i are not required to be distinct. Moreover, it can be several one-way roads between the same pair of cities.

Output

Print four integers — numbers of cities which Cicasso will visit according to optimal choice of the route. Numbers of cities should be printed in the order that Cicasso will visit them. If there are multiple solutions, print any of them.

Example

input

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

output

```
2 1 8 7
```

Note

Let $d(x, y)$ be the shortest distance between cities x and y. Then in the example $d(2, 1) = 3, d(1, 8) = 7, d(8, 7) = 3$. The total distance equals 13.

C. Codeword

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The famous sculptor Cicasso is a Reberlandian spy!

These is breaking news in Berlandian papers today. And now the sculptor is hiding. This time you give the shelter to the maestro. You have a protected bunker and you provide it to your friend. You set the security system in such way that only you can open the bunker. To open it one should solve the problem which is hard for others but is simple for you.

Every day the bunker generates a codeword s. Every time someone wants to enter the bunker, integer n appears on the screen. As the answer one should enter another integer — the residue modulo $10^9 + 7$ of the number of strings of length n that consist only of lowercase English letters and contain the string s as the **subsequence**.

The subsequence of string a is a string b that can be derived from the string a by removing some symbols from it (maybe none or all of them). In particular any string is the subsequence of itself. For example, the string "`cfo`" is the subsequence of the string "`codeforces`".

You haven't implemented the algorithm that calculates the correct answers yet and you should do that ASAP.

Input

The first line contains integer m ($1 \leq m \leq 10^5$) — the number of the events in the test case.

The second line contains nonempty string s — the string generated by the bunker for the current day.

The next m lines contain the description of the events. The description starts from integer t — the type of the event.

If $t = 1$ consider a new day has come and now a new string s is used. In that case the same line contains a new value of the string s.

If $t = 2$ integer n is given $(1 \leq n \leq 10^5)$. This event means that it's needed to find the answer for the current string s and the value n.

The sum of lengths of all generated strings doesn't exceed 10^5. All of the given strings consist only of lowercase English letters.

Output

For each query of the type 2 print the answer modulo $10^9 + 7$ on the separate line.

Example

input

```
3
a
2 2
1 bc
2 5
```

output

```
51
162626
```

Note

In the first event words of the form "a?" and "?a" are counted, where ? is an arbitrary symbol. There are 26 words of each of these types, but the word "aa" satisfies both patterns, so the answer is 51.

D. Chain Reaction

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Group of Berland scientists, with whom you have a close business relationship, makes a research in the area of peaceful nuclear energy. In particular, they found that a group of four nanobots, placed on a surface of a plate, can run a powerful chain reaction under certain conditions.

To be precise, researchers introduced a rectangular Cartesian coordinate system on a flat plate and selected four distinct points with integer coordinates where bots will be placed initially. Next each bot will be assigned with one of the four directions (up, down, left or right) parallel to the coordinate axes. After that, each bot is shifted by an integer distance (which may be different for different bots) along its direction. The chain reaction starts, if the bots are in the corners of a square with **positive area** with sides parallel to the coordinate axes. **Each corner of the square must contain one nanobot.** This reaction will be stronger, if bots spend less time to move. We can assume that bots move with unit speed. In other words, the lesser is the maximum length traveled by bot, the stronger is reaction.

Scientists have prepared a set of plates and selected starting position for the bots for each plate. Now they ask you to assign the direction for each bot to move after *landing* such that the maximum length traveled by bot is as small as possible.

Input

The first line contains an integer number t ($1 \le t \le 50$) — the number of plates.

t descriptions of plates follow. A description of each plate consists of four lines. Each line consists of a pair of integers numbers x_i, y_i ($-10^8 \le x_i, y_i \le 10^8$) — coordinates of the next bot. All bots are in different locations.

Note, though, the problem can include several records in one test, you can hack other people's submissions only with the test of one plate, i.e. parameter t in a hack test should be equal to 1.

Output

Print answers for all plates separately. First goes a single integer number in a separate line. If scientists have made an unfortunate mistake and nanobots are not able to form the desired square, print -1. Otherwise, print the minimum possible length of the longest bot's path.

If a solution exists, in the next four lines print two integer numbers — positions of each bot after moving. Print bots' positions in the order they are specified in the input data.

If there are multiple solution, you can print any of them.

Examples

input

```
2
1 1
1 -1
-1 1
-1 -1
1 1
2 2
4 4
6 6
```

output

```
0
1 1
1 -1
-1 1
-1 -1
-1
```

E. Forensic Examination

time limit per test: 6 seconds
memory limit per test: 768 megabytes
input: standard input
output: standard output

The country of Reberland is the archenemy of Berland. Recently the authorities of Berland arrested a Reberlandian spy who tried to bring the leaflets intended for agitational propaganda to Berland illegally . The most leaflets contain substrings of the Absolutely Inadmissible Swearword and maybe even the whole word.

Berland legal system uses the difficult algorithm in order to determine the guilt of the spy. The main part of this algorithm is the following procedure.

All the m leaflets that are brought by the spy are numbered from 1 to m. After that it's needed to get the answer to q queries of the following kind: "*In which leaflet in the segment of numbers $[l, r]$ the substring of the Absolutely Inadmissible Swearword $[p_l, p_r]$ occurs more often?*".

The expert wants you to automate that procedure because this time texts of leaflets are too long. Help him!

Input

The first line contains the string s ($1 \le |s| \le 5 \cdot 10^5$) — the Absolutely Inadmissible Swearword. The string s consists of only lowercase English letters.

The second line contains the only integer m ($1 \le m \le 5 \cdot 10^4$) — the number of texts of leaflets for expertise.

Each of the next m lines contains the only string t_i — the text of the i-th leaflet. The sum of lengths of all leaflet texts doesn't exceed $5 \cdot 10^4$. The text of the leaflets consists of only lowercase English letters.

The next line contains integer q ($1 \le q \le 5 \cdot 10^5$) — the number of queries for expertise.

Finally, each of the last q lines contains four integers l, r, p_l, p_r ($1 \le l \le r \le m, 1 \le p_l \le p_r \le |s|$), where $|s|$ is the length of the Absolutely Inadmissible Swearword.

Output

Print q lines. The i-th of them should contain two integers — the number of the text with the most occurences and the number of occurences of the substring $[p_l, p_r]$ of the string s. If there are several text numbers print the smallest one.

Examples

input

```
suffixtree
3
suffixtreesareawesome
cartesiantreeisworsethansegmenttree
nyeeheeheee
2
1 2 1 10
1 3 9 10
```

output

```
1 1
3 4
```