A. Newspaper Headline

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

A newspaper is published in Walrusland. Its heading is s_1, it consists of lowercase Latin letters. Fangy the little walrus wants to buy several such newspapers, cut out their headings, glue them one to another in order to get one big string. After that walrus erase several letters from this string in order to get a new word s_2. It is considered that when Fangy erases some letter, there's no whitespace formed instead of the letter. That is, the string remains unbroken and it still only consists of lowercase Latin letters.

For example, the heading is "abc". If we take two such headings and glue them one to the other one, we get "abcabc". If we erase the letters on positions 1 and 5, we get a word "bcac".

Which least number of newspaper headings s_1 will Fangy need to glue them, erase several letters and get word s_2?

Input

The input data contain two lines. The first line contain the heading s_1, the second line contains the word s_2. The lines only consist of lowercase Latin letters ($1 \le |s_1| \le 10^4, 1 \le |s_2| \le 10^6$).

Output

If it is impossible to get the word s_2 in the above-described manner, print "-1" (without the quotes). Otherwise, print the least number of newspaper headings s_1, which Fangy will need to receive the word s_2.

Examples

input

```
abc
xyz
```

output

```
-1
```

input

```
abcd
dabc
```

output

```
2
```

B. Queue

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

There are n walruses standing in a queue in an airport. They are numbered starting from the queue's tail: the 1-st walrus stands at the end of the queue and the n-th walrus stands at the beginning of the queue. The i-th walrus has the age equal to a_i.

The i-th walrus becomes displeased if there's a younger walrus standing in front of him, that is, if exists such j ($i < j$), that $a_i > a_j$. The *displeasure* of the i-th walrus is equal to the number of walruses between him and the furthest walrus ahead of him, which is younger than the i-th one. That is, the further that young walrus stands from him, the stronger the displeasure is.

The airport manager asked you to count for each of n walruses in the queue his displeasure.

Input

The first line contains an integer n ($2 \le n \le 10^5$) — the number of walruses in the queue. The second line contains integers a_i ($1 \le a_i \le 10^9$).

Note that some walruses can have the same age but for the displeasure to emerge the walrus that is closer to the head of the queue needs to be **strictly younger** than the other one.

Output

Print n numbers: if the i-th walrus is pleased with everything, print "-1" (without the quotes). Otherwise, print the i-th walrus's displeasure: the number of other walruses that stand between him and the furthest from him younger walrus.

Examples

input

```
6
10 8 5 3 50 45
```

output

```
2 1 0 -1 0 -1
```

input

```
7
10 4 6 3 2 8 15
```

output

```
4 2 1 0 -1 -1 -1
```

input

```
5
10 3 1 10 11
```

output

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

C. Ski Base

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

A ski base is planned to be built in Walrusland. Recently, however, the project is still in the constructing phase. A large land lot was chosen for the construction. It contains n ski junctions, numbered from 1 to n. Initially the junctions aren't connected in any way.

In the constructing process m bidirectional ski roads will be built. The roads are built one after another: first the road number 1 will be built, then the road number 2, and so on. The i-th road connects the junctions with numbers a_i and b_i.

Track is the route with the following properties:

- The route is closed, that is, it begins and ends in one and the same junction.
- The route contains at least one road.
- The route doesn't go on one road more than once, however it can visit any junction any number of times.

Let's consider the *ski base* as a non-empty set of roads that can be divided into one or more tracks so that exactly one track went along each road of the chosen set. Besides, each track can consist only of roads from the chosen set. Ski base doesn't have to be connected.

Two ski bases are considered different if they consist of different road sets.

After building each new road the Walrusland government wants to know the number of variants of choosing a ski base based on some subset of the already built roads. The government asks you to help them solve the given problem.

Input

The first line contains two integers n and m ($2 \le n \le 10^5, 1 \le m \le 10^5$). They represent the number of junctions and the number of roads correspondingly. Then on m lines follows the description of the roads in the order in which they were built. Each road is described by a pair of integers a_i and b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$) — the numbers of the connected junctions. There could be more than one road between a pair of junctions.

Output

Print m lines: the i-th line should represent the number of ways to build a ski base after the end of construction of the road number i. The numbers should be printed modulo 1000000009 ($10^9 + 9$).

Examples

input

```
3 4
1 3
2 3
1 2
1 2
```

output

```
0
0
1
3
```

Note

Let us have 3 junctions and 4 roads between the junctions have already been built (as after building all the roads in the sample): 1 and 3, 2 and 3, 2 roads between junctions 1 and 2. The land lot for the construction will look like this:

The land lot for the construction will look in the following way:

We can choose a subset of roads in three ways:

In the first and the second ways you can choose one path, for example, 1 - 2 - 3 - 1. In the first case you can choose one path 1 - 2 - 1.

D. Grocer's Problem

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

Yesterday was a fair in a supermarket's grocery section. There were n jars with spices on the fair. Before the event the jars were numbered from 1 to n from the left to the right. After the event the jars were moved and the grocer had to sort them by the increasing of the numbers.

The grocer has a special machine at his disposal. The machine can take any 5 or less jars and rearrange them in the way the grocer wants. Note that the jars **do not have to** stand consecutively. For example, from the permutation $2, 6, 5, 4, 3, 1$ one can get permutation $1, 2, 3, 4, 5, 6$, if pick the jars on the positions $1, 2, 3, 5$ and 6.

Which minimum number of such operations is needed to arrange all the jars in the order of their numbers' increasing?

Input

The first line contains an integer n ($1 \le n \le 10^5$). The second line contains n space-separated integers a_i ($1 \le a_i \le n$) — the i-th number represents the number of a jar that occupies the i-th position. It is guaranteed that all the numbers are distinct.

Output

Print on the first line the least number of operations needed to rearrange all the jars in the order of the numbers' increasing. Then print the description of all actions in the following format.

On the first line of the description of one action indicate the number of jars that need to be taken (k), on the second line indicate from which positions the jars need to be taken ($b_1, b_2, \dots, b_k$), on the third line indicate the jar's new order ($c_1, c_2, \dots, c_k$). After the operation is fulfilled the jar from position b_i will occupy the position c_i. The set ($c_1, c_2, \dots, c_k$) should be the rearrangement of the set ($b_1, b_2, \dots, b_k$).

If there are multiple solutions, output any.

Examples

input

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

output

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

input

```
14
9 13 11 3 10 7 12 14 1 5 4 6 8 2
```

output

```
3
4
2 13 8 14
13 8 14 2
5
6 7 12 5 10
7 12 6 10 5
5
3 11 4 1 9
11 4 3 9 1
```

Note

Let's consider the first sample. The jars can be sorted within two actions.

During the first action we take the jars from positions 1, 3, 6 and 4 and put them so that the jar that used to occupy the position 1 will occupy the position 3 after the operation is completed. The jar from position 3 will end up in position 6, the jar from position 6 will end up in position 4 and the jar from position 4 will end up in position 1.

After the first action the order will look like that: $1, 5, 3, 4, 2, 6$.

During the second operation the jars in positions 2 and 5 will change places.

E. Igloo Skyscraper

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

Today the North Pole hosts an Olympiad in a sport called... toy igloo skyscrapers' building!

There are n walruses taking part in the contest. Each walrus is given a unique number from 1 to n. After start each walrus begins to build his own igloo skyscraper. Initially, at the moment of time equal to 0, the height of the skyscraper i-th walrus is equal to a_i. Each minute the i-th walrus finishes building b_i floors.

The journalists that are reporting from the spot where the Olympiad is taking place, make q queries to the organizers. Each query is characterized by a group of three numbers l_i, r_i, t_i. The organizers respond to each query with a number x, such that:

1. Number x lies on the interval from l_i to r_i inclusive ($l_i \le x \le r_i$).

2. The skyscraper of the walrus number x possesses the maximum height among the skyscrapers of all walruses from the interval $[l_i, r_i]$ at the moment of time t_i.

For each journalists' query print the number of the walrus x that meets the above-given criteria. If there are several possible answers, print any of them.

Input

The first line contains numbers n and q ($1 \le n, q \le 10^5$). Next n lines contain pairs of numbers a_i, b_i ($1 \le a_i, b_i \le 10^9$). Then follow q queries i the following format l_i, r_i, t_i, one per each line ($1 \le l_i \le r_i \le n$, $0 \le t_i \le 10^6$). All input numbers are integers.

Output

For each journalists' query print the number of the walrus x that meets the criteria, given in the statement. Print one number per line.

Examples

input

```
5 4
4 1
3 5
6 2
3 5
6 5
1 5 2
1 3 5
1 1 0
1 5 0
```

output

```
5
2
1
5
```

input

```
5 4
6 1
5 1
2 5
4 3
6 1
2 4 1
3 4 5
1 4 5
1 2 0
```

output

```
3
3
3
1
```