Codeforces Beta Round #75 (Div. 2 Only)

A. Chips

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

There are n walruses sitting in a circle. All of them are numbered in the clockwise order: the walrus number 2 sits to the left of the walrus number 1, the walrus number 3 sits to the left of the walrus number 2, ..., the walrus number 1 sits to the left of the walrus number n.

The presenter has m chips. The presenter stands in the middle of the circle and starts giving the chips to the walruses starting from walrus number 1 and moving clockwise. The walrus number i gets i chips. If the presenter can't give the current walrus the required number of chips, then the presenter takes the remaining chips and the process ends. Determine by the given n and m how many chips the presenter will get in the end.

Input

The first line contains two integers n and m ($1 \le n \le 50$, $1 \le m \le 10^4$) — the number of walruses and the number of chips correspondingly.

Output

Print the number of chips the presenter ended up with.

Sample test(s)

input

```
4 11
```

output

```
0
```

input

```
17 107
```

output

```
2
```

input

```
3 8
```

output

```
1
```

Note

In the first sample the presenter gives one chip to the walrus number 1, two chips to the walrus number 2, three chips to the walrus number 3, four chips to the walrus number 4, then again one chip to the walrus number 1. After that the presenter runs out of chips. He can't give anything to the walrus number 2 and the process finishes.

In the third sample the presenter gives one chip to the walrus number 1, two chips to the walrus number 2, three chips to the walrus number 3, then again one chip to the walrus number 1. The presenter has one chip left and he can't give two chips to the walrus number 2, that's why the presenter takes the last chip.

B. Binary Number

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

Little walrus Fangy loves math very much. That's why when he is bored he plays with a number performing some operations.

Fangy takes some positive integer x and wants to get a number one from it. While x is not equal to 1, Fangy repeats the following action: if x is odd, then he adds 1 to it, otherwise he divides x by 2. Fangy knows that for any positive integer number the process ends in finite time.

How many actions should Fangy perform to get a number one from number x?

Input

The first line contains a positive integer x in a **binary system**. It is guaranteed that the first digit of x is different from a zero and the number of its digits does not exceed 10^6.

Output

Print the required number of actions.

Sample test(s)

input

```
1
```

output

```
0
```

input

```
1001001
```

output

```
12
```

input

```
101110
```

output

```
8
```

Note

Let's consider the third sample. Number 101110 is even, which means that we should divide it by 2. After the dividing Fangy gets an odd number 10111 and adds one to it. Number 11000 can be divided by 2 three times in a row and get number 11. All that's left is to increase the number by one (we get 100), and then divide it by 2 two times in a row. As a result, we get 1.

C. 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.

Sample test(s)

input

```
abc
xyz
```

output

```
-1
```

input

```
abcd
dabc
```

output

```
2
```

D. 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.

Sample test(s)

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
```

E. 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 ($10^9 + 9$).

Sample test(s)

input

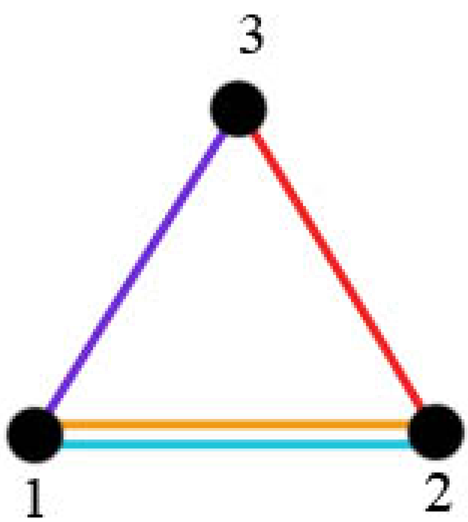
```
3 4
1 3
2 3
1 2
1 2
```

output

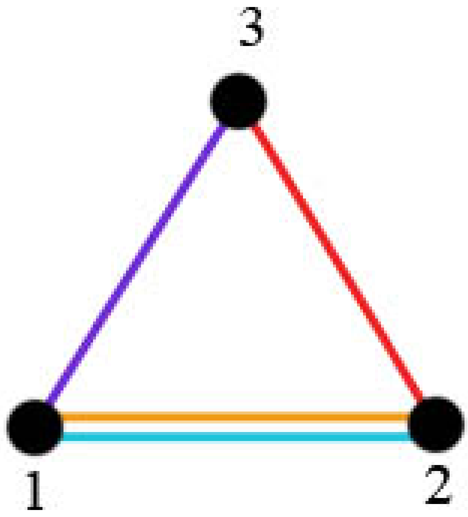
```
0
0
1
3
```

Note

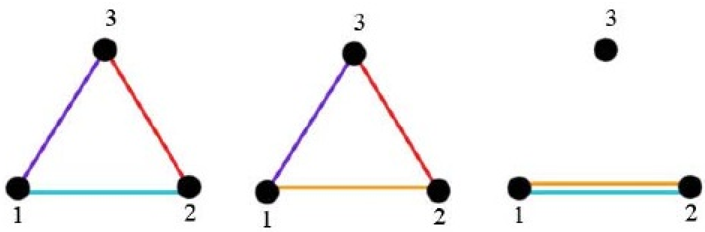
Let us have 3 junctions and 4 roads between the junctions have already been built (as after building all the roads in the sample): 1 and 3, 2 and 3, 2 roads between junctions 1 and 2. The land lot for the construction will look like this:



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.