ICM Technex 2017 and Codeforces Round #400 (Div. 1 + Div. 2, combined)

A. A Serial Killer

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

Our beloved detective, Sherlock is currently trying to catch a serial killer who kills a person each day. Using his powers of deduction, he came to know that the killer has a strategy for selecting his next victim.

The killer starts with two potential victims on his first day, selects one of these two, kills selected victim and replaces him with a new person. He repeats this procedure each day. This way, each day he has two potential victims to choose from. Sherlock knows the initial two potential victims. Also, he knows the murder that happened on a particular day and the new person who replaced this victim.

You need to help him get all the pairs of potential victims at each day so that Sherlock can observe some pattern.

Input

First line of input contains two names (length of each of them doesn't exceed 10), the two initials potential victims. Next line contains integer n ($1 \le n \le 1000$), the number of days.

Next n lines contains two names (length of each of them doesn't exceed 10), first being the person murdered on this day and the second being the one who replaced that person.

The input format is consistent, that is, a person murdered is guaranteed to be from the two potential victims at that time. Also, all the names are guaranteed to be distinct and consists of lowercase English letters.

Output

Output $n + 1$ lines, the i-th line should contain the two persons from which the killer selects for the i-th murder. The $(n + 1)$-th line should contain the two persons from which the next victim is selected. In each line, the two names can be printed in any order.

Examples

input

```
ross rachel
4
ross joey
rachel phoebe
phoebe monica
monica chandler
```

output

```
ross rachel
joey rachel
joey phoebe
joey monica
joey chandler
```

input

```
icm codeforces
1
codeforces technex
```

output

```
icm codeforces
icm technex
```

Note

In first example, the killer starts with `ross` and `rachel`.

- After day 1, `ross` is killed and `joey` appears.
- After day 2, `rachel` is killed and `phoebe` appears.
- After day 3, `phoebe` is killed and `monica` appears.
- After day 4, `monica` is killed and `chandler` appears.

B. Sherlock and his girlfriend

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

Sherlock has a new girlfriend (so unlike him!). Valentine's day is coming and he wants to gift her some jewelry.

He bought n pieces of jewelry. The i-th piece has price equal to $i + 1$, that is, the prices of the jewelry are $2, 3, 4, \ldots n + 1$.

Watson gave Sherlock a challenge to color these jewelry pieces such that two pieces don't have the same color if the price of one piece is a prime divisor of the price of the other piece. Also, Watson asked him to minimize the number of different colors used.

Help Sherlock complete this trivial task.

Input

The only line contains single integer n ($1 \le n \le 100000$) — the number of jewelry pieces.

Output

The first line of output should contain a single integer k, the minimum number of colors that can be used to color the pieces of jewelry with the given constraints.

The next line should consist of n space-separated integers (between 1 and k) that specify the color of each piece in the order of increasing price.

If there are multiple ways to color the pieces using k colors, you can output any of them.

Examples

input

```
3
```

output

```
2
1 1 2
```

input

```
4
```

output

```
2
2 1 1 2
```

Note

In the first input, the colors for first, second and third pieces of jewelry having respective prices 2, 3 and 4 are 1, 1 and 2 respectively.

In this case, as 2 is a prime divisor of 4, colors of jewelry having prices 2 and 4 must be distinct.

C. Molly's Chemicals

time limit per test: 2.5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Molly Hooper has n different kinds of chemicals arranged in a line. Each of the chemicals has an affection value, The i-th of them has affection value a_i.

Molly wants Sherlock to fall in love with her. She intends to do this by mixing a contiguous segment of chemicals together to make a love potion with total affection value as a non-negative **integer** power of k. Total affection value of a continuous segment of chemicals is the sum of affection values of each chemical in that segment.

Help her to do so in finding the total number of such segments.

Input

The first line of input contains two integers, n and k, the number of chemicals and the number, such that the total affection value is a non-negative power of this number k. ($1 \le n \le 10^5$, $1 \le |k| \le 10$).

Next line contains n integers $a_1, a_2, ..., a_n$ ($-10^9 \le a_i \le 10^9$) — affection values of chemicals.

Output

Output a single integer — the number of valid segments.

Examples

input

```
4 2
2 2 2 2
```

output

```
8
```

input

```
4 -3
3 -6 -3 12
```

output

```
3
```

Note

Do keep in mind that $k^0 = 1$.

In the first sample, Molly can get following different affection values:

- 2: segments $[1, 1]$, $[2, 2]$, $[3, 3]$, $[4, 4]$;
- 4: segments $[1, 2]$, $[2, 3]$, $[3, 4]$;
- 6: segments $[1, 3]$, $[2, 4]$;
- 8: segments $[1, 4]$.

Out of these, 2, 4 and 8 are powers of $k = 2$. Therefore, the answer is 8.

In the second sample, Molly can choose segments $[1, 2]$, $[3, 3]$, $[3, 4]$.

D. The Door Problem

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

Moriarty has trapped n people in n distinct rooms in a hotel. Some rooms are locked, others are unlocked. But, there is a condition that the people in the hotel can only escape when all the doors are unlocked at the same time. There are m switches. Each switch control doors of some rooms, but each door is controlled by **exactly two** switches.

You are given the initial configuration of the doors. Toggling any switch, that is, turning it ON when it is OFF, or turning it OFF when it is ON, toggles the condition of the doors that this switch controls. Say, we toggled switch 1, which was connected to room 1, 2 and 3 which were respectively locked, unlocked and unlocked. Then, after toggling the switch, they become unlocked, locked and locked.

You need to tell Sherlock, if there exists a way to unlock all doors at the same time.

Input

First line of input contains two integers n and m ($2 \le n \le 10^5$, $2 \le m \le 10^5$) — the number of rooms and the number of switches.

Next line contains n space-separated integers $r_1, r_2, ..., r_n$ ($0 \le r_i \le 1$) which tell the status of room doors. The i-th room is locked if $r_i = 0$, otherwise it is unlocked.

The i-th of next m lines contains an integer x_i ($0 \le x_i \le n$) followed by x_i distinct integers separated by space, denoting the number of rooms controlled by the i-th switch followed by the room numbers that this switch controls. It is guaranteed that the room numbers are in the range from 1 to n. It is guaranteed that each door is controlled by exactly two switches.

Output

Output "YES" without quotes, if it is possible to open all doors at the same time, otherwise output "NO" without quotes.

Examples

input

```
3 3
1 0 1
2 1 3
2 1 2
2 2 3
```

output

```
NO
```

input

```
3 3
1 0 1
3 1 2 3
1 2
2 1 3
```

output

```
YES
```

input

```
3 3
1 0 1
3 1 2 3
2 1 2
1 3
```

output

```
NO
```

Note

In the second example input, the initial statuses of the doors are $[1, 0, 1]$ (0 means locked, 1 — unlocked).

After toggling switch 3, we get $[0, 0, 0]$ that means all doors are locked.

Then, after toggling switch 1, we get $[1, 1, 1]$ that means all doors are unlocked.

It can be seen that for the first and for the third example inputs it is not possible to make all doors unlocked.

E. The Holmes Children

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

The Holmes children are fighting over who amongst them is the cleverest.

Mycroft asked Sherlock and Eurus to find value of $f(n)$, where $f(1) = 1$ and for $n \geq 2$, $f(n)$ is the number of distinct ordered positive integer pairs (x, y) that satisfy $x + y = n$ and $gcd(x, y) = 1$. The integer $gcd(a, b)$ is the greatest common divisor of a and b.

Sherlock said that solving this was child's play and asked Mycroft to instead get the value of . Summation is done over all positive integers d that divide n.

Eurus was quietly observing all this and finally came up with her problem to astonish both Sherlock and Mycroft.

She defined a k-composite function $F_k(n)$ recursively as follows:

She wants them to tell the value of $F_k(n)$ modulo 1000000007.

Input

A single line of input contains two space separated integers n ($1 \leq n \leq 10^{12}$) and k ($1 \leq k \leq 10^{12}$) indicating that Eurus asks Sherlock and Mycroft to find the value of $F_k(n)$ modulo 1000000007.

Output

Output a single integer — the value of $F_k(n)$ modulo 1000000007.

Examples

input

```
7 1
```

output

```
6
```

input

```
10 2
```

output

```
4
```

Note

In the first case, there are 6 distinct ordered pairs $(1, 6)$, $(2, 5)$, $(3, 4)$, $(4, 3)$, $(5, 2)$ and $(6, 1)$ satisfying $x + y = 7$ and $gcd(x, y) = 1$. Hence, $f(7) = 6$. So, $F_1(7) = f(g(7)) = f(f(7) + f(1)) = f(6 + 1) = f(7) = 6$.

F. Sherlock's bet to Moriarty

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Sherlock met Moriarty for a final battle of wits. He gave him a regular n sided convex polygon. In addition to it, he gave him certain diagonals to form regions on the polygon. It was guaranteed that the diagonals did not intersect in interior points.

He took each of the region and calculated its importance value. Importance value for a region formed by vertices $a_1, a_2, \ldots, a_x$ of the polygon will be given by $2^{a_1} + 2^{a_2} + \ldots + 2^{a_x}$. Then, he sorted these regions on the basis of their importance value in ascending order. After that he assigned each region an index from 1 to k, where k is the number of regions, and index of region is its position in the sorted array calculated above.

He wants Moriarty to color the regions using not more than 20 colors, such that two regions have same color only if all the simple paths between these two regions have at least one region with color value less than the color value assigned to these regions. Simple path between two regions f and h is a sequence of regions $r_1, r_2, \ldots r_t$ such that $r_1 = f$, $r_t = h$, for each $1 \le i < t$ regions r_i and r_{i+1} share an edge, and $r_i = r_j$ if and only if $i = j$.

Moriarty couldn't answer and asks Sherlock to solve it himself. Help Sherlock in doing so.

Input

First line contains two integers n and m ($3 \le n \le 100000$, $0 \le m \le n - 3$), the number of vertices in the polygon and the number of diagonals added.

Each of the next m lines contains two integers a and b ($1 \le a, b \le n$), describing a diagonal between vertices a and b. It is guaranteed that the diagonals are correct, i. e. a and b don't coincide and are not neighboring. It is guaranteed that the diagonals do not intersect.

Output

Let the number of regions be k.

Output k space-separated integers, each between 1 and 20, representing the colors of the regions in the order of increasing importance.

If there are multiple answers, print any of them. It can be shown that at least one answer exists.

Examples

input

```
4 1
1 3
```

output

```
1 2
```

input

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

output

```
2 1 2 3
```

Note

In 2nd input, regions formed in order after sorting will be $(1, 2, 3)$, $(1, 3, 4)$, $(1, 4, 5)$, $(1, 5, 6)$, i.e, region $(1, 2, 3)$ is first region followed by region $(1, 3, 4)$ and so on.

So, we can color regions 1 and 3 with same color, as region number 2 is on the path from 1 to 3 and it has color 1 which is less than color of 1 and 3, i.e., color number 2.

G. Sherlock and the Encrypted Data

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Sherlock found a piece of encrypted data which he thinks will be useful to catch Moriarty. The encrypted data consists of two integer l and r. He noticed that these integers were in hexadecimal form.

He takes each of the integers from l to r, and performs the following operations:

1. He lists the distinct digits present in the given number. For example: for 1014_{16}, he lists the digits as $1, 0, 4$.
2. Then he sums respective powers of two for each digit listed in the step above. Like in the above example $sum = 2^1 + 2^0 + 2^4 = 19_{10}$.
3. He changes the initial number by applying bitwise `xor` of the initial number and the sum. Example: . Note that `xor` is done in binary notation.

One more example: for integer `1e` the sum is $sum = 2^1 + 2^{14}$. Letters `a`, `b`, `c`, `d`, `e`, `f` denote hexadecimal digits 10, 11, 12, 13, 14, 15, respertively.

Sherlock wants to count the numbers in the range from l to r (both inclusive) which decrease on application of the above four steps. He wants you to answer his q queries for different l and r.

Input

First line contains the integer q ($1 \le q \le 10000$).

Each of the next q lines contain two hexadecimal integers l and r ($0 \le l \le r < 16^{15}$).

The hexadecimal integers are written using digits from 0 to 9 and/or lowercase English letters `a`, `b`, `c`, `d`, `e`, `f`.

The hexadecimal integers do not contain extra leading zeros.

Output

Output q lines, i-th line contains answer to the i-th query (in decimal notation).

Examples

input

```
1
1014 1014
```

output

```
1
```

input

```
2
1 1e
1 f
```

output

```
1
0
```

input

```
2
1 abc
d0e fe23
```

output

```
412
28464
```

Note

For the second input,

$$14_{16} = 20_{10}$$

$$sum = 2^1 + 2^4 = 18$$

Thus, it reduces. And, we can verify that it is the only number in range 1 to $1e$ that reduces.