A. Kuroni and the Gifts

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

Kuroni has n daughters. As gifts for them, he bought n necklaces and n bracelets:

- the i-th necklace has a brightness a_i, where all the a_i are **pairwise distinct** (i.e. all a_i are different),
- the i-th bracelet has a brightness b_i, where all the b_i are **pairwise distinct** (i.e. all b_i are different).

Kuroni wants to give **exactly one** necklace and **exactly one** bracelet to each of his daughters. To make sure that all of them look unique, the total brightnesses of the gifts given to each daughter should be **pairwise distinct**. Formally, if the i-th daughter receives a necklace with brightness x_i and a bracelet with brightness y_i, then the sums $x_i + y_i$ should be pairwise distinct. Help Kuroni to distribute the gifts.

For example, if the brightnesses are $a = [1, 7, 5]$ and $b = [6, 1, 2]$, then we may distribute the gifts as follows:

- Give the third necklace and the first bracelet to the first daughter, for a total brightness of $a_3 + b_1 = 11$.
- Give the first necklace and the third bracelet to the second daughter, for a total brightness of $a_1 + b_3 = 3$.
- Give the second necklace and the second bracelet to the third daughter, for a total brightness of $a_2 + b_2 = 8$.

Here is an example of an **invalid** distribution:

- Give the first necklace and the first bracelet to the first daughter, for a total brightness of $a_1 + b_1 = 7$.
- Give the second necklace and the second bracelet to the second daughter, for a total brightness of $a_2 + b_2 = 8$.
- Give the third necklace and the third bracelet to the third daughter, for a total brightness of $a_3 + b_3 = 7$.

This distribution is **invalid**, as the total brightnesses of the gifts received by the first and the third daughter are the same. Don't make them this upset!

Input

The input consists of multiple test cases. The first line contains an integer t $(1 \le t \le 100)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n $(1 \le n \le 100)$ — the number of daughters, necklaces and bracelets.

The second line of each test case contains n **distinct** integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 1000)$ — the brightnesses of the necklaces.

The third line of each test case contains n **distinct** integers $b_1, b_2, \dots, b_n$ $(1 \le b_i \le 1000)$ — the brightnesses of the bracelets.

Output

For each test case, print a line containing n integers $x_1, x_2, \dots, x_n$, representing that the i-th daughter receives a necklace with brightness x_i. In the next line print n integers $y_1, y_2, \dots, y_n$, representing that the i-th daughter receives a bracelet with brightness y_i.

The sums $x_1 + y_1, x_2 + y_2, \dots, x_n + y_n$ should all be distinct. The numbers $x_1, \dots, x_n$ should be equal to the numbers $a_1, \dots, a_n$ in some order, and the numbers $y_1, \dots, y_n$ should be equal to the numbers $b_1, \dots, b_n$ in some order.

It can be shown that an answer always exists. If there are multiple possible answers, you may print any of them.

Example

input

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

output

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

Note

In the first test case, it is enough to give the i-th necklace and the i-th bracelet to the i-th daughter. The corresponding sums are $1 + 8 = 9$, $8 + 4 = 12$, and $5 + 5 = 10$.

The second test case is described in the statement.

B. Kuroni and Simple Strings

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

Now that Kuroni has reached 10 years old, he is a big boy and doesn't like arrays of integers as presents anymore. This year he wants a Bracket sequence as a Birthday present. More specifically, he wants a bracket sequence so complex that no matter how hard he tries, he will not be able to remove a simple subsequence!

We say that a string formed by n characters '(' or ')' is **simple** if its length n is even and positive, its first $\frac{n}{2}$ characters are '(', and its last $\frac{n}{2}$ characters are ')'. For example, the strings () and (()) are simple, while the strings)(and ()() are not simple.

Kuroni will be given a string formed by characters '(' and ')' (the given string is not necessarily simple). An operation consists of choosing a subsequence of the characters of the string that forms a simple string and removing all the characters of this subsequence from the string. **Note that this subsequence doesn't have to be continuous**. For example, he can apply the operation to the string ')**()(**()**))**', to choose a subsequence of bold characters, as it forms a simple string '(())', delete these bold characters from the string and to get '))()'.

Kuroni has to perform the minimum possible number of operations on the string, in such a way that no more operations can be performed on the remaining string. The resulting string **does not** have to be empty.

Since the given string is too large, Kuroni is unable to figure out how to minimize the number of operations. Can you help him do it instead?

A sequence of characters a is a subsequence of a string b if a can be obtained from b by deletion of several (possibly, zero or all) characters.

Input

The only line of input contains a string s ($1 \le |s| \le 1000$) formed by characters '(' and ')', where $|s|$ is the length of s.

Output

In the first line, print an integer k — the minimum number of operations you have to apply. Then, print $2k$ lines describing the operations in the following format:

For each operation, print a line containing an integer m — the number of characters in the subsequence you will remove.

Then, print a line containing m integers $1 \le a_1 < a_2 < \dots < a_m$ — the indices of the characters you will remove. All integers must be less than or equal to the length of the current string, and the corresponding subsequence must form a simple string.

If there are multiple valid sequences of operations with the smallest k, you may print any of them.

Examples

input
```
(()((
```
output
```
1
2
1 3
```

input
```
)(
```
output
```
0
```

input
```
(()())
```
output
```
1
4
1 2 5 6
```

Note

In the first sample, the string is '**(()**(('. The operation described corresponds to deleting the bolded subsequence. The resulting string is '(((', and no more operations can be performed on it. Another valid answer is choosing indices 2 and 3, which results in the same final string.

In the second sample, it is already impossible to perform any operations.

C. Kuroni and Impossible Calculation

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

To become the king of Codeforces, Kuroni has to solve the following problem.

He is given n numbers $a_1, a_2, \dots, a_n$. Help Kuroni to calculate $\prod_{1\le i<j\le n} |a_i - a_j|$. As result can be very big, output it modulo m.

If you are not familiar with short notation, $\prod_{1\le i<j\le n} |a_i - a_j|$ is equal to $|a_1 - a_2|\cdot|a_1 - a_3|\cdot \dots \cdot|a_1 - a_n|\cdot|a_2 - a_3|\cdot|a_2 - a_4|\cdot \dots \cdot|a_2 - a_n| \cdot \dots \cdot |a_{n-1} - a_n|$. In other words, this is the product of $|a_i - a_j|$ for all $1\le i < j \le n$.

Input

The first line contains two integers n, m ($2\le n \le 2\cdot 10^5$, $1\le m \le 1000$) — number of numbers and modulo.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 10^9$).

Output

Output the single number — $\prod_{1\le i<j\le n} |a_i - a_j| \bmod m$.

Examples

input

```
2 10
8 5
```

output

```
3
```

input

```
3 12
1 4 5
```

output

```
0
```

input

```
3 7
1 4 9
```

output

```
1
```

Note

In the first sample, $|8 - 5| = 3 \equiv 3 \bmod 10$.

In the second sample, $|1 - 4|\cdot|1 - 5|\cdot|4 - 5| = 3\cdot 4 \cdot 1 = 12 \equiv 0 \bmod 12$.

In the third sample, $|1 - 4|\cdot|1 - 9|\cdot|4 - 9| = 3 \cdot 8 \cdot 5 = 120 \equiv 1 \bmod 7$.

D. Kuroni and the Celebration

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

This is an interactive problem.

After getting AC after 13 Time Limit Exceeded verdicts on a geometry problem, Kuroni went to an Italian restaurant to celebrate this holy achievement. Unfortunately, the excess sauce disoriented him, and he's now lost!

The United States of America can be modeled as a tree (why though) with n vertices. The tree is rooted at vertex r, wherein lies Kuroni's hotel.

Kuroni has a phone app designed to help him in such emergency cases. To use the app, he has to input two vertices u and v, and it'll return a vertex w, which is the lowest common ancestor of those two vertices.

However, since the phone's battery has been almost drained out from live-streaming Kuroni's celebration party, he could only use the app at most $\lfloor \frac{n}{2} \rfloor$ times. After that, the phone would die and there will be nothing left to help our dear friend! :(

As the night is cold and dark, Kuroni needs to get back, so that he can reunite with his comfy bed and pillow(s). Can you help him figure out his hotel's location?

Interaction

The interaction starts with reading a single integer n ($2 \le n \le 1000$), the number of vertices of the tree.

Then you will read $n-1$ lines, the i-th of them has two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$), denoting there is an edge connecting vertices x_i and y_i. It is guaranteed that the edges will form a tree.

Then you can make queries of type "`? u v`" ($1 \le u, v \le n$) to find the lowest common ancestor of vertex u and v.

After the query, read the result w as an integer.

In case your query is invalid or you asked more than $\lfloor \frac{n}{2} \rfloor$ queries, the program will print -1 and will finish interaction. You will receive a **Wrong answer** verdict. Make sure to exit immediately to avoid getting other verdicts.

When you find out the vertex r, print "`!` r" and quit after that. This query does not count towards the $\lfloor \frac{n}{2} \rfloor$ limit.

Note that the tree is fixed beforehand and will not change during the queries, i.e. the interactor is not adaptive.

After printing any query do not forget to print end of line and flush the output. Otherwise, you might get `Idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see the documentation for other languages.

Hacks

To hack, use the following format:

The first line should contain two integers n and r ($2 \le n \le 1000$, $1 \le r \le n$), denoting the number of vertices and the vertex with Kuroni's hotel.

The i-th of the next $n-1$ lines should contain two integers x_i and y_i ($1 \le x_i, y_i \le n$) — denoting there is an edge connecting vertex x_i and y_i.

The edges presented should form a tree.

Example

input

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

3

4

4
```

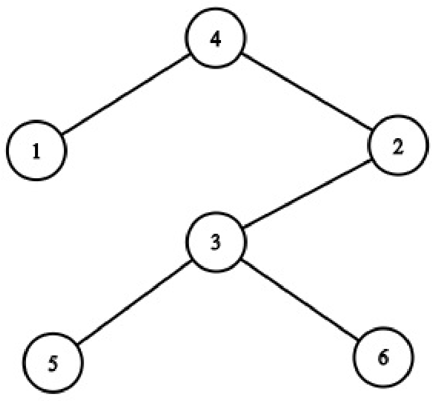
output

```






? 5 6

? 3 1

? 1 2

! 4
```

Note

Note that the example interaction contains extra empty lines so that it's easier to read. The real interaction doesn't contain any empty lines and you shouldn't print any extra empty lines as well.

The image below demonstrates the tree in the sample test:



E. Kuroni and the Score Distribution

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

Kuroni is the coordinator of the next Mathforces round written by the "Proof by AC" team. All the preparation has been done, and he is discussing with the team about the score distribution for the round.

The round consists of n problems, numbered from 1 to n. The problems are ordered in increasing order of difficulty, no two problems have the same difficulty. A score distribution for the round can be denoted by an array $a_1, a_2, \dots, a_n$, where a_i is the score of i-th problem.

Kuroni thinks that the score distribution should satisfy the following requirements:

- The score of each problem should be a positive integer not exceeding 10^9.
- A harder problem should grant a strictly higher score than an easier problem. In other words, $1 \le a_1 < a_2 < \dots < a_n \le 10^9$.
- The **balance** of the score distribution, defined as the number of triples (i, j, k) such that $1 \le i < j < k \le n$ and $a_i + a_j = a_k$, should be exactly m.

Help the team find a score distribution that satisfies Kuroni's requirement. In case such a score distribution does not exist, output -1.

Input

The first and single line contains two integers n and m ($1 \le n \le 5000$, $0 \le m \le 10^9$) — the number of problems and the required balance.

Output

If there is no solution, print a single integer -1.

Otherwise, print a line containing n integers $a_1, a_2, \dots, a_n$, representing a score distribution that satisfies all the requirements. If there are multiple answers, print any of them.

Examples

input

```
5 3
```

output

```
4 5 9 13 18
```

input

```
8 0
```

output

```
10 11 12 13 14 15 16 17
```

input

```
4 10
```

output

```
-1
```

Note

In the first example, there are 3 triples (i, j, k) that contribute to the balance of the score distribution.

- $(1, 2, 3)$
- $(1, 3, 4)$
- $(2, 4, 5)$

F. Kuroni and the Punishment

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Kuroni is very angry at the other setters for using him as a theme! As a punishment, he forced them to solve the following problem:

You have an array a consisting of n positive integers. An operation consists of choosing an element and either adding 1 to it or subtracting 1 from it, such that the element remains positive. We say the array is **good** if the greatest common divisor of all its elements is not 1. Find the minimum number of operations needed to make the array good.

Unable to match Kuroni's intellect, the setters failed to solve the problem. Help them escape from Kuroni's punishment!

Input

The first line contains an integer n $(2 \le n \le 2 \cdot 10^5)$ — the number of elements in the array.

The second line contains n integers $a_1, a_2, \dots, a_n$. $(1 \le a_i \le 10^{12})$ — the elements of the array.

Output

Print a single integer — the minimum number of operations required to make the array good.

Examples

input

```
3
6 2 4
```

output

```
0
```

input

```
5
9 8 7 3 1
```

output

```
4
```

Note

In the first example, the first array is already good, since the greatest common divisor of all the elements is 2.

In the second example, we may apply the following operations:

1. Add 1 to the second element, making it equal to 9.
2. Subtract 1 from the third element, making it equal to 6.
3. Add 1 to the fifth element, making it equal to 2.
4. Add 1 to the fifth element again, making it equal to 3.

The greatest common divisor of all elements will then be equal to 3, so the array will be good. It can be shown that no sequence of three or less operations can make the array good.

G. Kuroni and Antihype

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

Kuroni isn't good at economics. So he decided to found a new financial pyramid called **Antihype**. It has the following rules:

1. You can join the pyramid for free and get 0 coins.
2. If you are already a member of Antihype, you can invite your friend who is currently not a member of Antihype, and get a number of coins equal to your age (for each friend you invite).

n people have heard about Antihype recently, the i-th person's age is a_i. Some of them are friends, but friendship is a weird thing now: the i-th person is a friend of the j-th person **if and only if** $a_i \text{ AND } a_j = 0$, where AND denotes the bitwise AND operation.

Nobody among the n people is a member of Antihype at the moment. They want to cooperate to join and invite each other to

Antihype in a way that maximizes their combined gainings. Could you help them?

Input

The first line contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of people.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 2 \cdot 10^5)$ — the ages of the people.

Output

Output exactly one integer — the maximum possible combined gainings of all n people.

Example

input

```
3
1 2 3
```

output

```
2
```

Note

Only the first and second persons are friends. The second can join Antihype and invite the first one, getting 2 for it.

H. Kuroni the Private Tutor

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

As a professional private tutor, Kuroni has to gather statistics of an exam. Kuroni has appointed you to complete this important task. You must not disappoint him.

The exam consists of n questions, and m students have taken the exam. Each question was worth 1 point. Question i was solved by at least l_i and at most r_i students. Additionally, you know that the total score of all students is t.

Furthermore, you took a glance at the final ranklist of the quiz. The students were ranked from 1 to m, where rank 1 has the highest score and rank m has the lowest score. Ties were broken arbitrarily.

You know that the student at rank p_i had a score of s_i for $1 \le i \le q$.

You wonder if there could have been a huge tie for first place. Help Kuroni determine the maximum number of students who could have gotten as many points as the student with rank 1, and the maximum possible score for rank 1 achieving this maximum number of students.

Input

The first line of input contains two integers $(1 \le n, m \le 10^5)$, denoting the number of questions of the exam and the number of students respectively.

The next n lines contain two integers each, with the i-th line containing l_i and r_i $(0 \le l_i \le r_i \le m)$.

The next line contains a single integer q $(0 \le q \le m)$.

The next q lines contain two integers each, denoting p_i and s_i $(1 \le p_i \le m, 0 \le s_i \le n)$. It is guaranteed that all p_i are distinct and if $p_i \le p_j$, then $s_i \ge s_j$.

The last line contains a single integer t $(0 \le t \le nm)$, denoting the total score of all students.

Output

Output two integers: the maximum number of students who could have gotten as many points as the student with rank 1, and the maximum possible score for rank 1 achieving this maximum number of students. If there is no valid arrangement that fits the given data, output -1 -1.

Examples

input

```
5 4
2 4
2 3
1 1
0 1
0 0
1
4 1
7
```

output

```
3 2
```

input

```
5 6
```

```
0 6
0 6
2 5
6 6
4 6
1
3 3
30
```

output

```
-1 -1
```

Note

For the first sample, here is one possible arrangement that fits the data:

Students 1 and 2 both solved problems 1 and 2.

Student 3 solved problems 2 and 3.

Student 4 solved problem 4.

The total score of all students is $T = 7$. Note that the scores of the students are 2, 2, 2 and 1 respectively, which satisfies the condition that the student at rank 4 gets exactly 1 point. Finally, 3 students tied for first with a maximum score of 2, and it can be proven that we cannot do better with any other arrangement.