A. EhAb AnD gCd

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

You are given a positive integer x. Find **any** such 2 positive integers a and b such that $GCD(a,b)+LCM(a,b)=x$.

As a reminder, $GCD(a,b)$ is the greatest integer that divides both a and b. Similarly, $LCM(a,b)$ is the smallest integer such that both a and b divide it.

It's guaranteed that the solution always exists. If there are several such pairs (a, b), you can output any of them.

Input

The first line contains a single integer t $(1 \le t \le 100)$ — the number of testcases.

Each testcase consists of one line containing a single integer, x $(2 \le x \le 10^9)$.

Output

For each testcase, output a pair of positive integers a and b ($1 \le a, b \le 10^9)$ such that $GCD(a,b)+LCM(a,b)=x$. It's guaranteed that the solution always exists. If there are several such pairs (a, b), you can output any of them.

Example

input

```
2
2
14
```

output

```
1 1
6 4
```

Note

In the first testcase of the sample, $GCD(1,1)+LCM(1,1)=1+1=2$.

In the second testcase of the sample, $GCD(6,4)+LCM(6,4)=2+12=14$.

B. CopyCopyCopyCopyCopy

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

Ehab has an array a of length n. He has just enough free time to make a new array consisting of n copies of the old array, written back-to-back. What will be the length of the new array's longest increasing subsequence?

A sequence a is a subsequence of an array b if a can be obtained from b by deletion of several (possibly, zero or all) elements. The longest increasing subsequence of an array is the longest subsequence such that its elements are ordered in strictly increasing order.

Input

The first line contains an integer t — the number of test cases you need to solve. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 10^5$) — the number of elements in the array a.

The second line contains n space-separated integers a_1, a_2, $\ldots$, a_n ($1 \le a_i \le 10^9$) — the elements of the array a.

The sum of n across the test cases doesn't exceed 10^5.

Output

For each testcase, output the length of the longest increasing subsequence of a if you concatenate it to itself n times.

Example

input

```
2
3
```

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

output

```
3
5
```

Note

In the first sample, the new array is $[3, 2, \mathbf{1}, 3, \mathbf{2}, 1, \mathbf{3}, 2, 1]$. The longest increasing subsequence is marked in bold.

In the second sample, the longest increasing subsequence will be $[1, 3, 4, 5, 9]$.

C. Ehab and Path-etic MEXs

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

You are given a tree consisting of n nodes. You want to write some labels on the tree's edges such that the following conditions hold:

- Every label is an integer between 0 and $n-2$ inclusive.
- All the written labels are distinct.
- The largest value among $MEX(u, v)$ over all pairs of nodes (u, v) is as small as possible.

Here, $MEX(u, v)$ denotes the smallest non-negative integer that isn't written on any edge on the unique simple path from node u to node v.

Input

The first line contains the integer n ($2 \le n \le 10^5$) — the number of nodes in the tree.

Each of the next $n-1$ lines contains two space-separated integers u and v ($1 \le u, v \le n$) that mean there's an edge between nodes u and v. It's guaranteed that the given graph is a tree.

Output

Output $n-1$ integers. The i^{th} of them will be the number written on the i^{th} edge (in the input order).

Examples

input

```
3
1 2
1 3
```

output

```
0
1
```

input

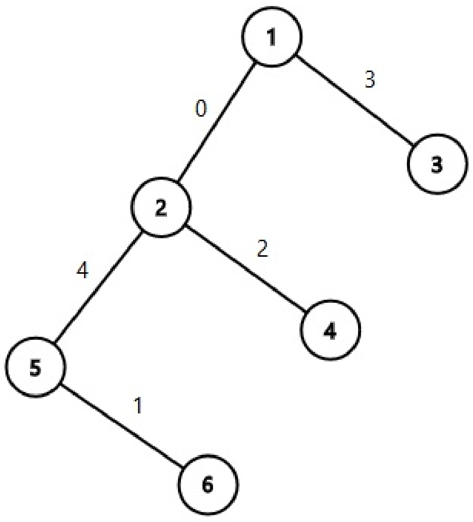
```
6
1 2
1 3
2 4
2 5
5 6
```

output

```
0
3
2
4
1
```

Note

The tree from the second sample:



D. Ehab the Xorcist

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

Given 2 integers u and v, find the shortest array such that bitwise-xor of its elements is u, and the sum of its elements is v.

Input

The only line contains 2 integers u and v $(0 \le u, v \le 10^{18})$.

Output

If there's no array that satisfies the condition, print "-1". Otherwise:

The first line should contain one integer, n, representing the length of the desired array. The next line should contain n **positive** integers, the array itself. If there are multiple possible answers, print any.

Examples

input
```
2 4
```
output
```
2
3 1
```

input
```
1 3
```
output
```
3
1 1 1
```

input
```
8 5
```
output
```
-1
```

input
```
0 0
```
output
```
0
```

Note

In the first sample, $3 \oplus 1 = 2$ and $3 + 1 = 4$. There is no valid array of smaller length.

Notice that in the fourth sample the array is empty.

E. Ehab's REAL Number Theory Problem

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

You are given an array a of length n that has a special condition: every element in this array has at most 7 divisors. Find the length of the shortest non-empty subsequence of this array product of whose elements is a perfect square.

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

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the length of a.

The second line contains n integers a_1, a_2, $\ldots$, a_n ($1 \le a_i \le 10^6$) — the elements of the array a.

Output

Output the length of the shortest non-empty subsequence of a product of whose elements is a perfect square. If there are several shortest subsequences, you can find any of them. If there's no such subsequence, print "-1".

Examples

input

```
3
1 4 6
```

output

```
1
```

input

```
4
2 3 6 6
```

output

```
2
```

input

```
3
6 15 10
```

output

```
3
```

input

```
4
2 3 5 7
```

output

```
-1
```

Note

In the first sample, you can choose a subsequence $[1]$.

In the second sample, you can choose a subsequence $[6,6]$.

In the third sample, you can choose a subsequence $[6,15,10]$.

In the fourth sample, there is no such subsequence.

F. Ehab's Last Theorem

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

It's the year 5555. You have a graph, and you want to find a long cycle and a huge independent set, just because you can. But for now, let's just stick with finding either.

Given a connected graph with n vertices, you can choose to either:

- find an independent set that has **exactly** $\lceil\sqrt{n}\rceil$ vertices.
- find a **simple** cycle of length **at least** $\lceil\sqrt{n}\rceil$.

An independent set is a set of vertices such that no two of them are connected by an edge. A simple cycle is a cycle that doesn't contain any vertex twice. I have a proof you can always solve one of these problems, but it's too long to fit this margin.

Input

The first line contains two integers n and m ($5 \le n \le 10^5$, $n-1 \le m \le 2 \cdot 10^5$) — the number of vertices and edges in the graph.

Each of the next m lines contains two space-separated integers u and v ($1 \le u, v \le n$) that mean there's an edge between vertices u and v. It's guaranteed that the graph is connected and doesn't contain any self-loops or multiple edges.

Output

If you choose to solve the first problem, then on the first line print "1", followed by a line containing $\lceil\sqrt{n}\rceil$ distinct integers not exceeding n, the vertices in the desired independent set.

If you, however, choose to solve the second problem, then on the first line print "2", followed by a line containing one integer, c, representing the length of the found cycle, followed by a line containing c distinct integers integers not exceeding n, the vertices in the desired cycle, **in the order they appear in the cycle**.

Examples

input

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

output

```
1
1 6 4
```

input

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

output

```
2
4
1 5 2 4
```

input

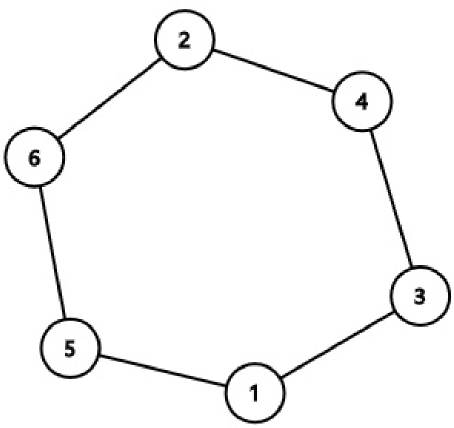
```
5 4
1 2
1 3
2 4
2 5
```

output

```
1
3 4 5
```

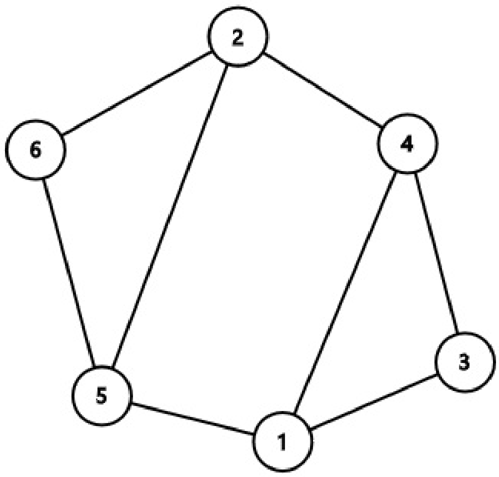
Note

In the first sample:



Notice that you can solve either problem, so printing the cycle $2-4-3-1-5-6$ is also acceptable.

In the second sample:



Notice that if there are multiple answers you can print any, so printing the cycle $2-5-6$, for example, is acceptable.

In the third sample: