A. Silent Classroom

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

There are n students in the first grade of Nlogonia high school. The principal wishes to split the students into two classrooms (each student must be in exactly one of the classrooms). Two distinct students whose name starts with the same letter will be chatty if they are put in the same classroom (because they must have a lot in common). Let x be the number of such pairs of students in a split. Pairs (a, b) and (b, a) are the same and counted only once.

For example, if there are 6 students: "olivia", "jacob", "tanya", "jack", "oliver" and "jessica", then:

- splitting into two classrooms ("jack", "jacob", "jessica", "tanya") and ("olivia", "oliver") will give $x = 4$ (3 chatting pairs in the first classroom, 1 chatting pair in the second classroom),
- splitting into two classrooms ("jack", "tanya", "olivia") and ("jessica", "oliver", "jacob") will give $x = 1$ (0 chatting pairs in the first classroom, 1 chatting pair in the second classroom).

You are given the list of the n names. What is the minimum x we can obtain by splitting the students into classrooms?

Note that it is valid to place all of the students in one of the classrooms, leaving the other one empty.

Input

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

After this n lines follow.

The i-th line contains the name of the i-th student.

It is guaranteed each name is a string of lowercase English letters of length at most 20. Note that multiple students may share the same name.

Output

The output must consist of a single integer x — the minimum possible number of chatty pairs.

Examples

input

```
4
jorge
jose
oscar
jerry
```

output

```
1
```

input

```
7
kambei
gorobei
shichiroji
kyuzo
heihachi
katsushiro
kikuchiyo
```

output

```
2
```

input

```
5
mike
mike
mike
mike
mike
```

output

```
4
```

Note

In the first sample the minimum number of pairs is 1. This can be achieved, for example, by putting everyone except jose in one classroom, and jose in the other, so jorge and jerry form the only chatty pair.

In the second sample the minimum number of pairs is 2. This can be achieved, for example, by putting kambei, gorobei, shichiroji and kyuzo in one room and putting heihachi, katsushiro and kikuchiyo in the other room. In this case the two pairs are kambei and kyuzo, and katsushiro and kikuchiyo.

In the third sample the minimum number of pairs is 4. This can be achieved by placing three of the students named mike in one classroom and the other two students in another classroom. Thus there will be three chatty pairs in one classroom and one chatty pair in the other classroom.

B. All the Vowels Please

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

Tom loves vowels, and he likes long words with many vowels. His favorite words are vowelly words. We say a word of length k is vowelly if there are positive integers n and m such that $n \cdot m = k$ and when the word is written by using n rows and m columns (the first row is filled first, then the second and so on, with each row filled from left to right), every vowel of the English alphabet appears at least once in every row and every column.

You are given an integer k and you must either print a vowelly word of length k or print -1 if no such word exists.

In this problem the vowels of the English alphabet are 'a', 'e', 'i', 'o' ,'u'.

Input

Input consists of a single line containing the integer k ($1 \le k \le 10^4$) — the required length.

Output

The output must consist of a single line, consisting of a vowelly word of length k consisting of lowercase English letters if it exists or

-1 if it does not.

If there are multiple possible words, you may output any of them.

Examples

input
7

output
-1

input
36

output
agoeuioaeiruuimaeoieauoweouoiaouimae

Note

In the second example, the word "agoeuioaeiruuimaeoieauoweouoiaouimae" can be arranged into the following 6×6 grid:

a	g	o	e	u	i
o	a	e	i	r	u
u	i	m	a	e	o
i	e	a	u	o	w
e	o	u	o	i	a
o	u	i	m	a	e

It is easy to verify that every row and every column contain all the vowels.

C. A Tale of Two Lands

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

The legend of the foundation of Vectorland talks of two integers x and y. Centuries ago, the array king placed two markers at points $|x|$ and $|y|$ on the number line and conquered all the land in between (including the endpoints), which he declared to be Arrayland. Many years later, the vector king placed markers at points $|x-y|$ and $|x+y|$ and conquered all the land in between (including the endpoints), which he declared to be Vectorland. He did so in such a way that the land of Arrayland was completely inside (including the endpoints) the land of Vectorland.

Here $|z|$ denotes the absolute value of z.

Now, Jose is stuck on a question of his history exam: "What are the values of x and y?" Jose doesn't know the answer, but he believes he has narrowed the possible answers down to n integers $a_1, a_2, \dots, a_n$. Now, he wants to know the number of **unordered** pairs formed by two **different** elements from these n integers such that the legend could be true if x and y were equal to these two values. Note that it is possible that Jose is wrong, and that no pairs could possibly make the legend true.

Input

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

The second line contains n pairwise distinct integers $a_1, a_2, \dots, a_n$ ($-10^9 \le a_i \le 10^9$) — the choices Jose is considering.

Output

Print a single integer number — the number of unordered pairs $\{x, y\}$ formed by different numbers from Jose's choices that could make the legend true.

Examples

input
3 2 5 -3

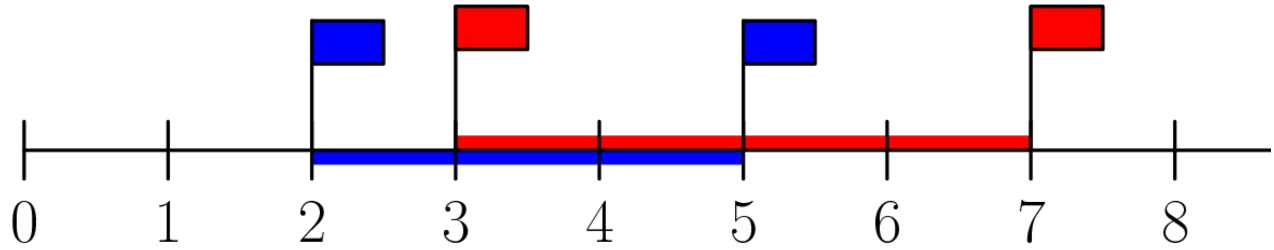
output
2

input
2 3 6

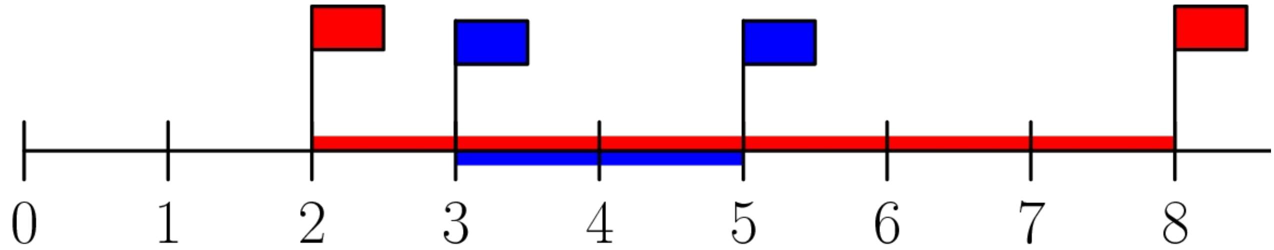
output
1

Note

Consider the first sample. For the pair $\{2, 5\}$, the situation looks as follows, with the Arrayland markers at $|2| = 2$ and $|5| = 5$, while the Vectorland markers are located at $|2-5| = 3$ and $|2+5| = 7$:

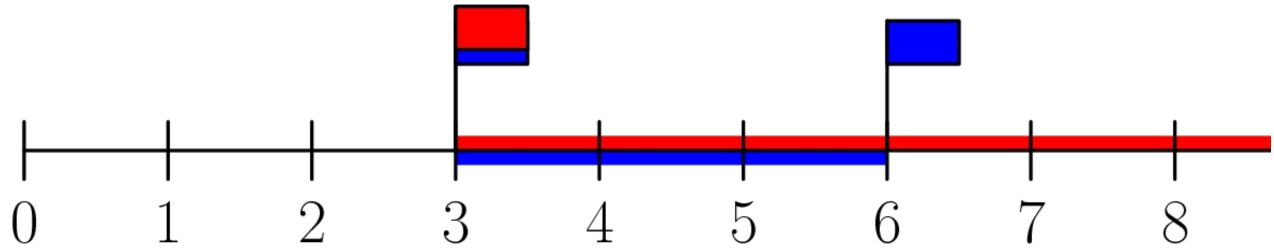


The legend is not true in this case, because the interval $[2, 3]$ is not conquered by Vectorland. For the pair $\{5, -3\}$ the situation looks as follows, with Arrayland consisting of the interval $[3, 5]$ and Vectorland consisting of the interval $[2, 8]$:



As Vectorland completely contains Arrayland, the legend is true. It can also be shown that the legend is true for the pair $\{2, -3\}$, for a total of two pairs.

In the second sample, the only pair is $\{3, 6\}$, and the situation looks as follows:



Note that even though Arrayland and Vectorland share 3 as endpoint, we still consider Arrayland to be completely inside of Vectorland.

D. Cute Sequences

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

Given a positive integer m, we say that a sequence $x_1, x_2, \dots, x_n$ of positive integers is m-cute if for every index i such that $2 \le i \le n$ it holds that $x_i = x_{i-1} + x_{i-2} + \dots + x_1 + r_i$ for some positive integer r_i satisfying $1 \le r_i \le m$.

You will be given q queries consisting of three positive integers a, b and m. For each query you must determine whether or not there exists an m-cute sequence whose first term is a and whose last term is b. If such a sequence exists, you must additionally find an example of it.

Input

The first line contains an integer number q ($1 \le q \le 10^3$) — the number of queries.

Each of the following q lines contains three integers a, b, and m ($1 \le a, b, m \le 10^{14}$, $a \le b$), describing a single query.

Output

For each query, if no m-cute sequence whose first term is a and whose last term is b exists, print -1.

Otherwise print an integer k ($1 \le k \le 50$), followed by k integers $x_1, x_2, \dots, x_k$ ($1 \le x_i \le 10^{14}$). These integers must satisfy $x_1 = a$, $x_k = b$, and that the sequence $x_1, x_2, \dots, x_k$ is m-cute.

It can be shown that under the problem constraints, for each query either no m-cute sequence exists, or there exists one with at most 50 terms.

If there are multiple possible sequences, you may print any of them.

Example

input

```
2
5 26 2
3 9 1
```

output

```
4 5 6 13 26
-1
```

Note

Consider the sample. In the first query, the sequence $5, 6, 13, 26$ is valid since $6 = 5 + \mathbf{1}$, $13 = 6 + 5 + \mathbf{2}$ and $26 = 13 + 6 + 5 + \mathbf{2}$ have the bold values all between 1 and 2, so the sequence is 2-cute. Other valid sequences, such as $5, 7, 13, 26$ are also accepted.

In the second query, the only possible 1-cute sequence starting at 3 is $3, 4, 8, 16, \dots$, which does not contain 9.

E. The LCMs Must be Large

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

Dora the explorer has decided to use her money after several years of juicy royalties to go shopping. What better place to shop than Nlogonia?

There are n stores numbered from 1 to n in Nlogonia. The i-th of these stores offers a **positive integer** a_i.

Each day among the last m days Dora bought a single integer from some of the stores. The same day, Swiper the fox bought a single integer from all the stores that Dora did not buy an integer from on that day.

Dora considers Swiper to be her rival, and she considers that she beat Swiper on day i if and only if the least common multiple of the numbers she bought on day i is strictly greater than the least common multiple of the numbers that Swiper bought on day i.

The least common multiple (LCM) of a collection of integers is the smallest positive integer that is divisible by all the integers in the collection.

However, Dora forgot the values of a_i. Help Dora find out if there are positive integer values of a_i such that she beat Swiper on **every** day. You don't need to find what are the possible values of a_i though.

Note that it is possible for some values of a_i to coincide in a solution.

Input

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

After this m lines follow, the i-th line starts with an integer s_i ($1 \le s_i \le n-1$), the number of integers Dora bought on day i, followed by s_i distinct integers, the indices of the stores where Dora bought an integer on the i-th day. The indices are between 1 and n.

Output

Output must consist of a single line containing "`possible`" if there exist positive integers a_i such that for each day the least common multiple of the integers bought by Dora is strictly greater than the least common multiple of the integers bought by Swiper on that day. Otherwise, print "`impossible`".

Note that you don't have to restore the integers themselves.

Examples

input

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

output

```
possible
```

input

```
10 10
1 1
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
```

output

```
impossible
```

Note

In the first sample, a possible choice for the values of the a_i is $3, 4, 3, 5, 2$. On the first day, Dora buys the integers $3, 4$ and 3, whose LCM is 12, while Swiper buys integers 5 and 2, whose LCM is 10. On the second day, Dora buys $3, 5$ and 2, whose LCM is 30, and Swiper buys integers 3 and 4, whose LCM is 12.

F. Vicky's Delivery Service

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In a magical land there are n cities conveniently numbered $1, 2, \dots, n$. Some pairs of these cities are connected by magical colored roads. Magic is unstable, so at any time, new roads may appear between two cities.

Vicky the witch has been tasked with performing deliveries between some pairs of cities. However, Vicky is a beginner, so she can only complete a delivery if she can move from her starting city to her destination city through a *double rainbow*. A double rainbow is a sequence of cities $c_1, c_2, \dots, c_k$ satisfying the following properties:

- For each i with $1 \le i \le k-1$, the cities c_i and c_{i+1} are connected by a road.
- For each i with $1 \le i \le \frac{k-1}{2}$, the roads connecting c_{2i} with c_{2i-1} and c_{2i+1} have the same color.

For example if $k=5$, the road between c_1 and c_2 must be the same color as the road between c_2 and c_3, and the road between c_3 and c_4 must be the same color as the road between c_4 and c_5.

Vicky has a list of events in chronological order, where each event is either a delivery she must perform, or appearance of a new road. Help her determine which of her deliveries she will be able to complete.

Input

The first line contains four integers n, m, c, and q ($2 \le n \le 10^5$, $1 \le m, c, q \le 10^5$), denoting respectively the number of cities, the number of roads initially present, the number of different colors the roads can take, and the number of events.

Each of the following m lines contains three integers x, y, and z ($1 \le x, y \le n$, $1 \le z \le c$), describing that there initially exists a bidirectional road with color z between cities x and y.

Then q lines follow, describing the events. Each event is one of the following two types:

1. `+ x y z` ($1 \le x, y \le n$, $1 \le z \le c$), meaning a road with color z appears between cities x and y;
2. `? x y` ($1 \le x, y \le n$), meaning you should determine whether Vicky can make a delivery starting at city x and ending at city y. It is guaranteed that $x \ne y$.

It is guaranteed that at any moment, there is at most one road connecting any pair of cities, and that no road connects a city to itself. It is guaranteed that the input contains at least one event of the second type.

Output

For each event of the second type, print a single line containing "`Yes`" (without quotes) if the delivery can be made, or a single line containing "`No`" (without quotes) otherwise.

Example

input

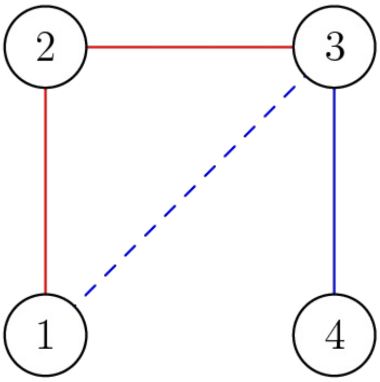
```
4 3 2 4
1 2 1
2 3 1
3 4 2
? 1 4
? 4 1
+ 3 1 2
? 4 1
```

output

```
Yes
No
Yes
```

Note

The following picture corresponds to the sample.



For her first delivery, Vicky can use the sequence `1, 2, 3, 4` which is a double rainbow. However, she cannot complete the second delivery, as she can only reach city 3. After adding the road between cities 1 and 3, she can now complete a delivery from city 4 to city 1 by using the double rainbow `4, 3, 1`.