A. NEKO's Maze Game

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

3R2 as DJ Mashiro - Happiness Breeze
Ice - DJ Mashiro is dead or alive

NEKO#ΦωΦ has just got a new maze game on her PC!

The game's main puzzle is a maze, in the forms of a $2 \times$ rectangle grid. NEKO's task is to lead a Nekomimi girl from cell $(1, 1)$ to the gate at $(2, \ )$ and escape the maze. The girl can only move between cells sharing a common side.

However, at some moments during the game, some cells may change their state: either from normal ground to lava (which forbids movement into that cell), or vice versa (which makes that cell passable again). Initially all cells are of the ground type.

After hours of streaming, NEKO finally figured out there are only such moments: the -th moment toggles the state of cell $(\ , \ )$ (either from ground to lava or vice versa).

Knowing this, NEKO wonders, after each of the moments, whether it is still possible to move from cell $(1, 1)$ to cell $(2, \ )$ without going through any lava cells.

Although NEKO is a great streamer and gamer, she still can't get through quizzes and problems requiring large amount of Brain Power. Can you help her?

Input

The first line contains integers , $(2 \le \quad \le 10^5, 1 \le \quad \le 10^5)$.

The -th of following lines contains two integers , $(1 \le \quad \le 2, 1 \le \quad \le \ )$, denoting the coordinates of the cell to be flipped at the -th moment.

It is guaranteed that cells $(1, 1)$ and $(2, \ )$ never appear in the query list.

Output

For each moment, if it is possible to travel from cell $(1, 1)$ to cell $(2, \ )$, print "Yes", otherwise print "No". There should be exactly answers, one after every update.

You can print the words in any case (either lowercase, uppercase or mixed).

Example

input

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

output

```
Yes
No
No
No
Yes
```

Note

We'll crack down the example test here:

- After the first query, the girl still able to reach the goal. One of the shortest path ways should be: $(1, 1) \to (1, 2) \to (1, 3) \to (1, 4) \to (1, 5) \to (2, 5)$.
- After the second query, it's impossible to move to the goal, since the farthest cell she could reach is $(1, 3)$.
- After the fourth query, the $(2, 3)$ is not blocked, but now all the 4-th column is blocked, so she still can't reach the goal.
- After the fifth query, the column barrier has been lifted, thus she can go to the final goal again.

B. Aroma's Search

time limit per test: 1 second
memory limit per test: 256 megabytes

input: standard input
output: standard output

THE SxPLAY & KIVΛ -
KIVΛ & Nikki Simmons - Perspectives

With a new body, our idol Aroma White (or should we call her Kaori Minamiya?) begins to uncover her lost past through the OS space.

The space can be considered a 2D plane, with an infinite number of data nodes, indexed from 0, with their coordinates defined as follows:

- The coordinates of the 0-th node is $({}_0,\ {}_0)$
- For $\ > 0$, the coordinates of -th node is $(\ \cdot\ {}_{-1} +\ ,\ \ \cdot\ {}_{-1} +\ )$

Initially Aroma stands at the point $(\ ,\ )$. She can stay in OS space for at most seconds, because after this time she has to warp back to the real world. She **doesn't** need to return to the entry point $(\ ,\ )$ to warp home.

While within the OS space, Aroma can do the following actions:

- From the point $(\ ,\ )$, Aroma can move to one of the following points: $(\ -1,\ )$, $(\ +1,\ )$, $(\ ,\ -1)$ or $(\ ,\ +1)$. This action requires 1 second.
- If there is a data node at where Aroma is staying, she can collect it. We can assume this action costs 0 seconds. Of course, each data node can be collected at most once.

Aroma wants to collect as many data as possible before warping back. Can you help her in calculating the maximum number of data nodes she could collect within seconds?

Input

The first line contains integers ${}_0,\ {}_0,\ ,\ ,\ ,\ $ $(1 \le\ {}_0,\ {}_0 \le 10^{16}, 2 \le\ ,\ \le 100, 0 \le\ ,\ \le 10^{16})$, which define the coordinates of the data nodes.

The second line contains integers $\ ,\ ,\ $ $(1 \le\ ,\ ,\ \le 10^{16})$ – the initial Aroma's coordinates and the amount of time available.

Output

Print a single integer — the maximum number of data nodes Aroma can collect within seconds.

Examples

input

```
1 1 2 3 1 0
2 4 20
```

output

```
3
```

input

```
1 1 2 3 1 0
15 27 26
```

output

```
2
```

input

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

output

```
0
```

Note

In all three examples, the coordinates of the first 5 data nodes are $(1,1)$, $(3,3)$, $(7,9)$, $(15,27)$ and $(31,81)$ (remember that nodes are numbered from 0).

In the first example, the optimal route to collect 3 nodes is as follows:

- Go to the coordinates $(3,3)$ and collect the 1-st node. This takes $|3-2|+|3-4|=2$ seconds.
- Go to the coordinates $(1,1)$ and collect the 0-th node. This takes $|1-3|+|1-3|=4$ seconds.
- Go to the coordinates $(7,9)$ and collect the 2-nd node. This takes $|7-1|+|9-1|=14$ seconds.

In the second example, the optimal route to collect 2 nodes is as follows:

- Collect the 3-rd node. This requires no seconds.
- Go to the coordinates $(7,9)$ and collect the 2-th node. This takes $|15-7|+|27-9|=26$ seconds.

In the third example, Aroma can't collect any nodes. She should have taken proper rest instead of rushing into the OS space like that.

C. Xenon's Attack on the Gangs

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

INSPION FullBand Master - INSPION
INSPION - IOLITE-SUNSTONE

On another floor of the A.R.C. Markland-N, the young man Simon "Xenon" Jackson, takes a break after finishing his project early (as always). Having a lot of free time, he decides to put on his legendary hacker "X" instinct and fight against the gangs of the cyber world.

His target is a network of small gangs. This network contains exactly -1 direct links, each of them connecting two gangs together. The links are placed in such a way that every pair of gangs is connected through a sequence of direct links.

By mining data, Xenon figured out that the gangs used a form of cross-encryption to avoid being busted: every link was assigned an integer from 0 to -2 such that all assigned integers are distinct and every integer was assigned to some link. If an intruder tries to access the encrypted data, they will have to surpass password layers, with being defined by the following formula:

$$= \sum_{1\le \; < \; \le} \quad (\; , \;)$$

Here, $(\; , \;)$ denotes the smallest non-negative integer that does not appear on any link on the unique simple path from gang to gang .

Xenon doesn't know the way the integers are assigned, but it's not a problem. He decides to let his AI's instances try all the passwords on his behalf, but before that, he needs to know the maximum possible value of , so that the AIs can be deployed efficiently.

Now, Xenon is out to write the AI scripts, and he is expected to finish them in two hours. Can you find the maximum possible before he returns?

Input

The first line contains an integer $(2 \le \quad \le 3000)$, the number of gangs in the network.

Each of the next -1 lines contains integers and $(1 \le \quad , \quad \le \quad ; \quad \ne \quad)$, indicating there's a direct link between gangs and .

It's guaranteed that links are placed in such a way that each pair of gangs will be connected by exactly one simple path.

Output

Print the maximum possible value of — the number of password layers in the gangs' network.

Examples

input

```
3
1 2
2 3
```

output

```
3
```

input

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

output

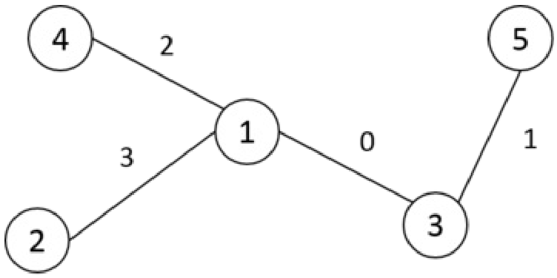
```
10
```

Note

In the first example, one can achieve the maximum with the following assignment:

With this assignment, $(1,2)=0$, $(1,3)=2$ and $(2,3)=1$. Therefore, $=0+2+1=3$.

In the second example, one can achieve the maximum with the following assignment:



With this assignment, all non-zero mex value are listed below:

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

Therefore, $= 1 + 2 + 1 + 2 + 1 + 3 = 10$.

D. Chaotic V.

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

Æsir - CHAOS
Æsir - V.

"Everything has been planned out. No more hidden concerns. The condition of Cytus is also perfect.

The time right now...... 00:01:12......

It's time."

The emotion samples are now sufficient. After almost 3 years, it's time for Ivy to awake her bonded sister, Vanessa.

The system inside A.R.C.'s Library core can be considered as an undirected graph with infinite number of processing nodes, numbered with all positive integers $(1, 2, 3, \ldots)$. The node with a number (> 1), is directly connected with a node with number $\frac{\quad}{(\ )}$, with $(\ )$ being the lowest prime divisor of .

Vanessa's mind is divided into fragments. Due to more than 500 years of coma, the fragments have been scattered: the -th fragment is now located at the node with a number $!$ (a factorial of).

To maximize the chance of successful awakening, Ivy decides to place the samples in a node , so that the total length of paths from each fragment to is smallest possible. If there are multiple fragments located at the same node, the path from that node to needs to be counted multiple times.

In the world of zeros and ones, such a requirement is very simple for Ivy. Not longer than a second later, she has already figured out such a node.

But for a mere human like you, is this still possible?

For simplicity, please answer the minimal sum of paths' lengths from every fragment to the emotion samples' assembly node .

Input

The first line contains an integer $(1 \leq \quad \leq 10^6)$ — number of fragments of Vanessa's mind.

The second line contains integers: $_1, \ _2, \ldots, \quad (0 \leq \quad \leq 5000)$, denoting the nodes where fragments of Vanessa's mind are located: the -th fragment is at the node with a number $!$.

Output

Print a single integer, denoting the minimal sum of path from every fragment to the node with the emotion samples (a.k.a. node).

As a reminder, if there are multiple fragments at the same node, the distance from that node to needs to be counted multiple times as well.

Examples

input
3 2 1 4

output
5

input

```
4
3 1 4 4
```

output

```
6
```

input

```
4
3 1 4 1
```

output

```
6
```

input

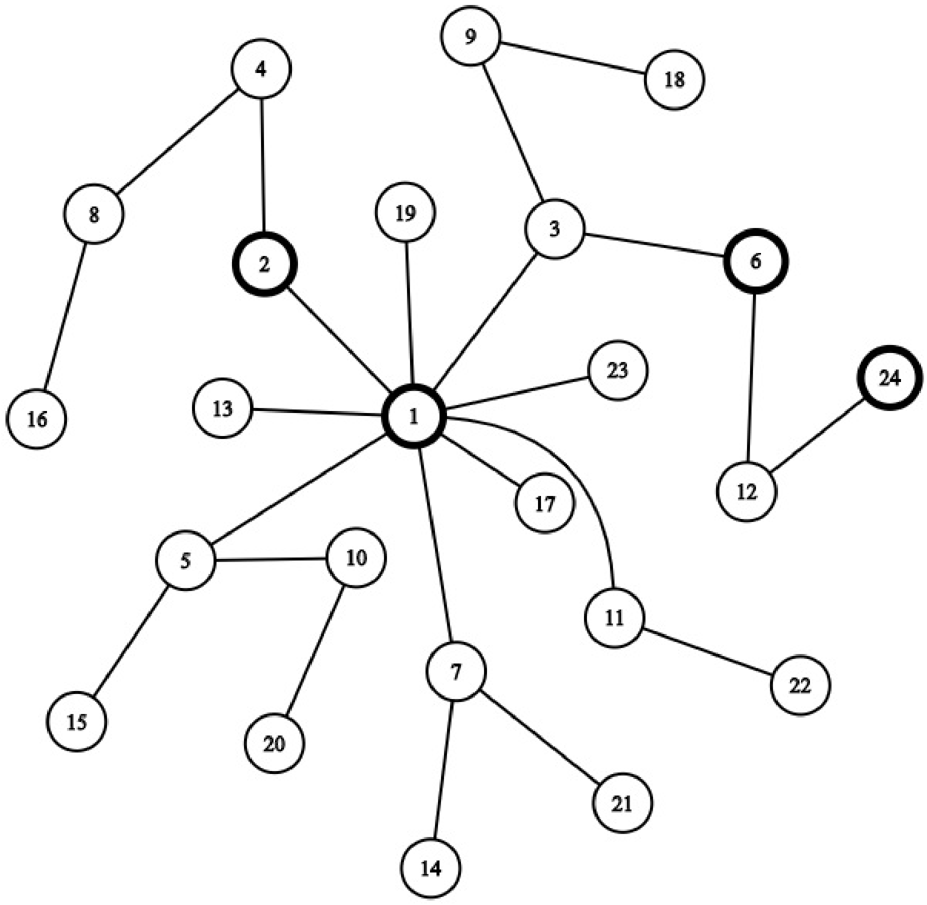
```
5
3 1 4 1 5
```

output

```
11
```

Note

Considering the first 24 nodes of the system, the node network will look as follows (the nodes $1!$, $2!$, $3!$, $4!$ are drawn bold):



For the first example, Ivy will place the emotion samples at the node 1. From here:

- The distance from Vanessa's first fragment to the node 1 is 1.
- The distance from Vanessa's second fragment to the node 1 is 0.
- The distance from Vanessa's third fragment to the node 1 is 4.

The total length is 5.

For the second example, the assembly node will be 6. From here:

- The distance from Vanessa's first fragment to the node 6 is 0.
- The distance from Vanessa's second fragment to the node 6 is 2.
- The distance from Vanessa's third fragment to the node 6 is 2.
- The distance from Vanessa's fourth fragment to the node 6 is again 2.

The total path length is 6.

E. Rin and The Unknown Flower

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

MisoilePunch♪ -

This is an interactive problem!

On a normal day at the hidden office in A.R.C. Markland-N, Rin received an artifact, given to her by the exploration captain Sagar.

After much analysis, she now realizes that this artifact contains data about a strange flower, which has existed way before the New Age. However, the information about its chemical structure has been encrypted heavily.

The chemical structure of this flower can be represented as a string . From the unencrypted papers included, Rin already knows the length of that string, and she can also conclude that the string contains at most three distinct letters: "C" (as in Carbon), "H" (as in Hydrogen), and "O" (as in Oxygen).

At each moment, Rin can input a string of an arbitrary length into the artifact's terminal, and it will return every starting position of as a *substring* of .

However, the artifact has limited energy and cannot be recharged in any way, since the technology is way too ancient and is incompatible with any current A.R.C.'s devices. To be specific:

- The artifact only contains $\frac{7}{5}$ units of energy.
- For each time Rin inputs a string of length , the artifact consumes $\frac{1}{2}$ units of energy.
- If the amount of energy reaches below zero, the task will be considered failed immediately, as the artifact will go black forever.

Since the artifact is so precious yet fragile, Rin is very nervous to attempt to crack the final data. Can you give her a helping hand?

Interaction

The interaction starts with a single integer $(1 \le \quad \le 500)$, the number of test cases. The interaction for each testcase is described below:

First, read an integer $(4 \le \quad \le 50)$, the length of the string .

Then you can make queries of type "? s" $(1 \le | \ | \le \ )$ to find the occurrences of as a substring of .

After the query, you need to read its result as a series of integers in a line:

- The first integer denotes the number of occurrences of as a substring of $(-1 \le \quad \le \ )$. If $= -1$, it means you have exceeded the energy limit or printed an invalid query, and you need to terminate immediately, to guarantee a "`Wrong answer`" verdict, otherwise you might get an arbitrary verdict because your solution will continue to read from a closed stream.
- The following integers $_1, \ _2, \ldots,$ $(1 \le \ _1 < \ _2 < \ldots < \quad \le \ )$ denote the starting positions of the substrings that match the string .

When you find out the string , print "! " to finish a test case. This query doesn't consume any energy. The interactor will return an integer 1 or 0. If the interactor returns 1, you can proceed to the next test case, or terminate the program if it was the last testcase.

If the interactor returns 0, it means that your guess is incorrect, and you should to terminate to guarantee a "`Wrong answer`" verdict.

Note that in every test case the string 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

For hack, use the following format. Note that you can only hack with one test case:

The first line should contain a single integer $(\ = 1)$.

The second line should contain an integer $(4 \le \quad \le 50)$ — the string's size.

The third line should contain a string of size , consisting of characters "C", "H" and "O" only. This is the string contestants will have to find out.

Examples

input

```
1
4

2 1 2

1 2

0

1
```

output

```


? C

? CH

? CCHO

! CCHH
```

input

```
2
5

0

2 2 3

1
8

1 5

1 5

1 3

2 1 2

1
```

output

```


? O

? HHH

! CHHHH


? COO

? COOH

? HCCOO

? HH

! HHHCCOOH
```

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.

F. Nora's Toy Boxes

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

SIHanatsuka - EMber
SIHanatsuka - ATONEMENT

Back in time, the seven-year-old Nora used to play lots of games with her creation ROBO_Head-02, both to have fun and enhance his abilities.

One day, Nora's adoptive father, Phoenix Wyle, brought Nora boxes of toys. Before unpacking, Nora decided to make a fun game for ROBO.

She labelled all boxes with distinct integers $_1, \ _2, \ldots,$ and asked ROBO to do the following action several (possibly zero) times:

- Pick three distinct indices , and , such that | and | . In other words, divides both and , that is $\bmod = 0$, $\bmod = 0$.
- After choosing, Nora will give the -th box to ROBO, and he will place it on top of the box pile at his side. Initially, the pile is empty.
- After doing so, the box becomes unavailable for any further actions.

Being amused after nine different tries of the game, Nora asked ROBO to calculate the number of possible different piles having the **largest** amount of boxes in them. Two piles are considered different if there exists a position where those two piles have different boxes.

Since ROBO was still in his infant stages, and Nora was still too young to concentrate for a long time, both fell asleep before finding the final answer. Can you help them?

As the number of such piles can be very large, you should print the answer modulo $10^9 + 7$.

Input

The first line contains an integer ($3 \leq \ \leq 60$), denoting the number of boxes.

The second line contains distinct integers $_1, \ _2, \ldots,$ ($1 \leq \ \leq 60$), where is the label of the -th box.

Output

Print the number of distinct piles having the maximum number of boxes that ROBO_Head can have, modulo $10^9 + 7$.

Examples

input

```
3
2 6 8
```

output

```
2
```

input

```
5
2 3 4 9 12
```

output

```
4
```

input

```
4
5 7 2 9
```

output

```
1
```

Note

Let's illustrate the box pile as a sequence , with the pile's bottommost box being at the leftmost position.

In the first example, there are 2 distinct piles possible:

- $= [6]$ ($[2, \ , 8] \xrightarrow{(1,3,2)} [2, 8]$)
- $= [8]$ ($[2, 6, \ ] \xrightarrow{(1,2,3)} [2, 6]$)

In the second example, there are 4 distinct piles possible:

- $= [9, 12]$ ($[2, 3, 4, \ , 12] \xrightarrow{(2,5,4)} [2, 3, 4, \ ] \xrightarrow{(1,3,4)} [2, 3, 4]$)
- $= [4, 12]$ ($[2, 3, \ , 9, 12] \xrightarrow{(1,5,3)} [2, 3, 9, \ ] \xrightarrow{(2,3,4)} [2, 3, 9]$)
- $= [4, 9]$ ($[2, 3, \ , 9, 12] \xrightarrow{(1,5,3)} [2, 3, \ , 12] \xrightarrow{(2,4,3)} [2, 3, 12]$)
- $= [9, 4]$ ($[2, 3, 4, \ , 12] \xrightarrow{(2,5,4)} [2, 3, \ , 12] \xrightarrow{(1,4,3)} [2, 3, 12]$)

In the third sequence, ROBO can do nothing at all. Therefore, there is only 1 valid pile, and that pile is empty.