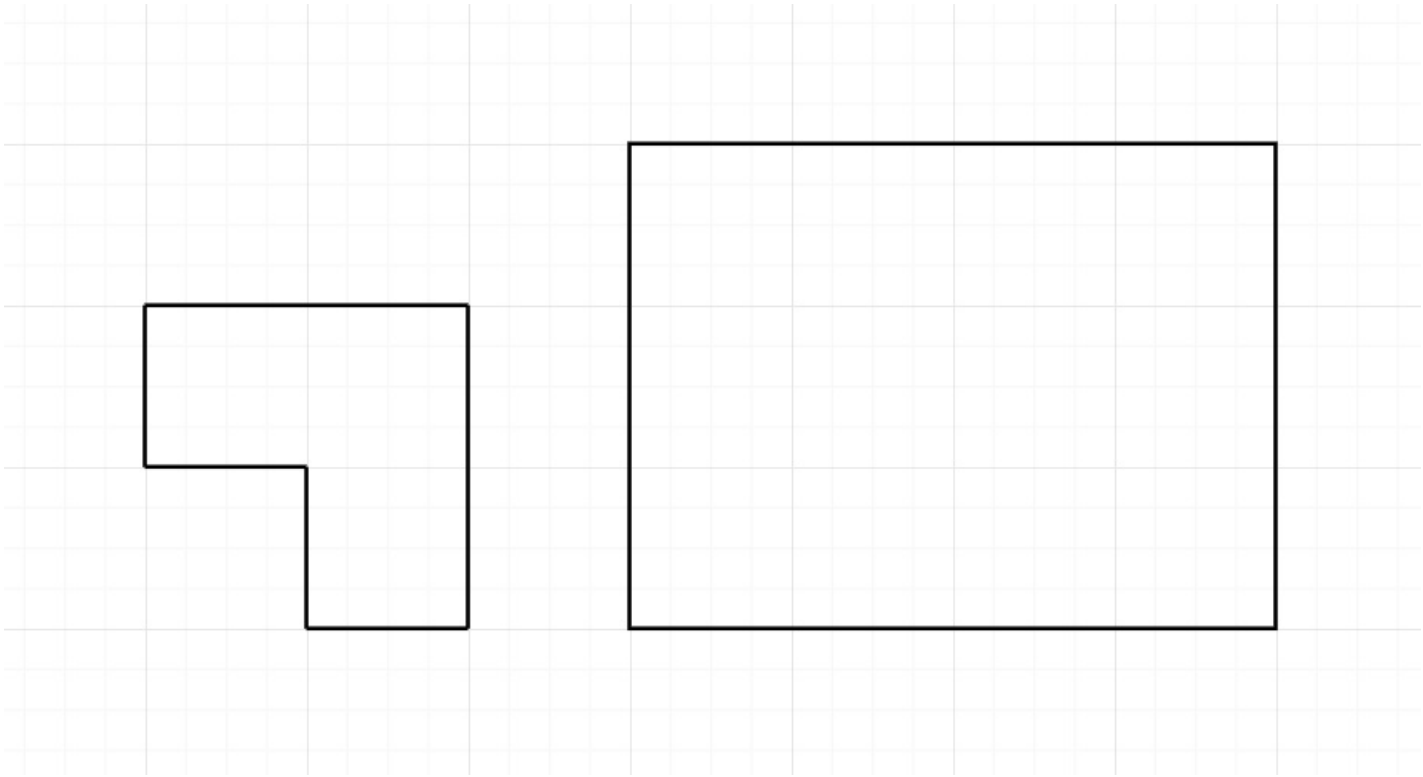
Codeforces Round #566 (Div. 2)

A. Filling Shapes

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

You have a given integer n. Find the number of ways to fill all $3 \times n$ tiles with the shape described in the picture below. Upon filling, no empty spaces are allowed. Shapes cannot overlap.



This picture describes when $n = 4$. The left one is the shape and the right one is $3 \times n$ tiles.

Input
The only line contains one integer n ($1 \leq n \leq 60$) — the length.

Output
Print the number of ways to fill.

Examples

input
4
output
4

input
1
output
0

Note
In the first example, there are 4 possible cases of filling.

In the second example, you cannot fill the shapes in 3×1 tiles.

B. Plus from Picture

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

output: standard output

You have a given picture with size $w \times h$. Determine if the given picture has a single "+" shape or not. A "+" shape is described below:

- A "+" shape has one center nonempty cell.
- There should be some (at least one) consecutive non-empty cells in each direction (left, right, up, down) from the center. In other words, there should be a ray in each direction.
- All other cells are empty.

Find out if the given picture has single "+" shape.

Input

The first line contains two integers h and w ($1 \leq h, w \leq 500$) — the height and width of the picture.

The i-th of the next h lines contains string s_i of length w consisting "`.`" and "*" where "`.`" denotes the empty space and "*" denotes the non-empty space.

Output

If the given picture satisfies all conditions, print "YES". Otherwise, print "NO".

You can output each letter in any case (upper or lower).

Examples

input
5 6 ....*... .****. ..*... ..*...
output
YES

input
3 5 ..*.. ****. .*...
output
NO

input
7 7 ....*... ..****. ...*... ...*... ....*.....
output
NO

input
5 6 ..**.. ..**.. ****** ..**.. ..**..
output
NO

input
3 7 .*...*. ***.*** .*...*.
output
NO

input
5 10

```
..........
...*.......
..*.******.
...*.......
..........
```

output
NO

Note

In the first example, the given picture contains one "+".

In the second example, two vertical branches are located in a different column.

In the third example, there is a dot outside of the shape.

In the fourth example, the width of the two vertical branches is 2.

In the fifth example, there are two shapes.

In the sixth example, there is an empty space inside of the shape.

C. Beautiful Lyrics

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

You are given n words, each of which consists of lowercase alphabet letters. Each word **contains at least** one vowel. You are going to choose some of the given words and make as many beautiful lyrics as possible.

Each *lyric* consists of two lines. Each *line* consists of two words separated by whitespace.

A lyric is *beautiful* if and only if it satisfies all conditions below.

- The number of vowels in the first word of the first line is the same as the number of vowels in the first word of the second line.
- The number of vowels in the second word of the first line is the same as the number of vowels in the second word of the second line.
- The last vowel of the first line is the same as the last vowel of the second line. Note that there may be consonants after the vowel.

Also, letters "a", "e", "o", "i", and "u" are vowels. Note that "y" is **never** vowel.

For example of a beautiful lyric,

```
"hello hellooowww"
"whatsup yowowowow"
```

is a beautiful lyric because there are two vowels each in "hello" and "whatsup", four vowels each in "hellooowww" and "yowowowow" (keep in mind that "y" is not a vowel), and the last vowel of each line is "o".
For example of a not beautiful lyric,

```
"hey man"
"iam mcdic"
```

is not a beautiful lyric because "hey" and "iam" don't have same number of vowels and the last vowels of two lines are different ("a" in the first and "i" in the second).
How many beautiful lyrics can you write from given words? Note that you cannot use a word more times than it is given to you. For example, if a word is given three times, you can use it at most three times.

Input

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

The i-th of the next n lines contains string s_i consisting lowercase alphabet letters — the i-th word. It is guaranteed that the sum of the total word length is equal or less than 10^6. Each word contains at least one vowel.

Output

In the first line, print m — the number of maximum possible beautiful lyrics.

In next $2m$ lines, print m beautiful lyrics (two lines per lyric).

If there are multiple answers, print any.

Examples

input
14 wow this

is the first mcdics codeforces round hooray i am proud about that
output
3 about proud hooray round wow first this is i that mcdics am

input
7 arsijo suggested the idea for this problem
output
0

input
4 same same same differ
output
1 same differ same same

Note

In the first example, those beautiful lyrics are one of the possible answers. Let's look at the first lyric on the sample output of the first example. "`about proud hooray round`" forms a beautiful lyric because "`about`" and "`hooray`" have same number of vowels, "`proud`" and "`round`" have same number of vowels, and both lines have same last vowel. On the other hand, you cannot form any beautiful lyric with the word "`codeforces`".

In the second example, you cannot form any beautiful lyric from given words.

In the third example, you can use the word "`same`" up to three times.

D. Complete Mirror

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

You have given tree consist of n vertices. Select a vertex as root vertex that satisfies the condition below.

- For all vertices v_1 and v_2, if $distance(root, v_1) = distance(root, v_2)$ then $degree(v_1) = degree(v_2)$, where $degree$ means the number of vertices connected to that vertex, and $distance$ means the number of edges between two vertices.

Determine and find if there is such root vertex in the tree. If there are multiple answers, find any of them.

Input

The first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of vertices.

Each of the next $n-1$ lines contains two integers v_i and u_i ($1 \leq v_i < u_i \leq n$) — it means there is an edge exist between v_i and u_i. It is guaranteed that the graph forms tree.

Output

If there is such root vertex exists, print any of them. Otherwise, print -1.

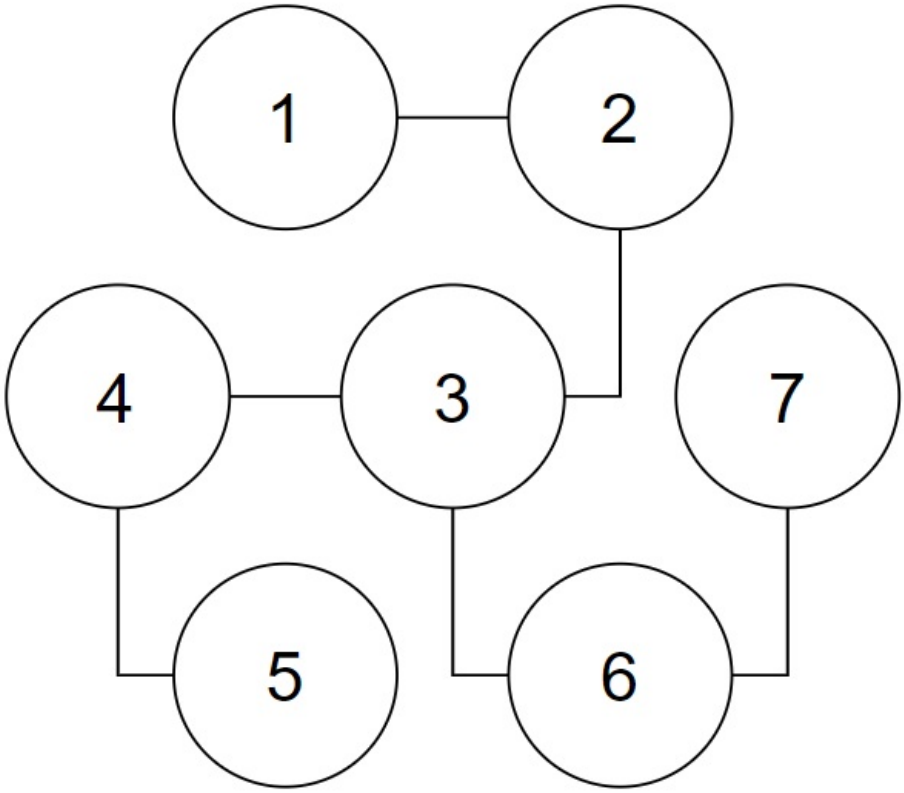
Examples

input
7 1 2 2 3 3 4 4 5 3 6 6 7
output
3

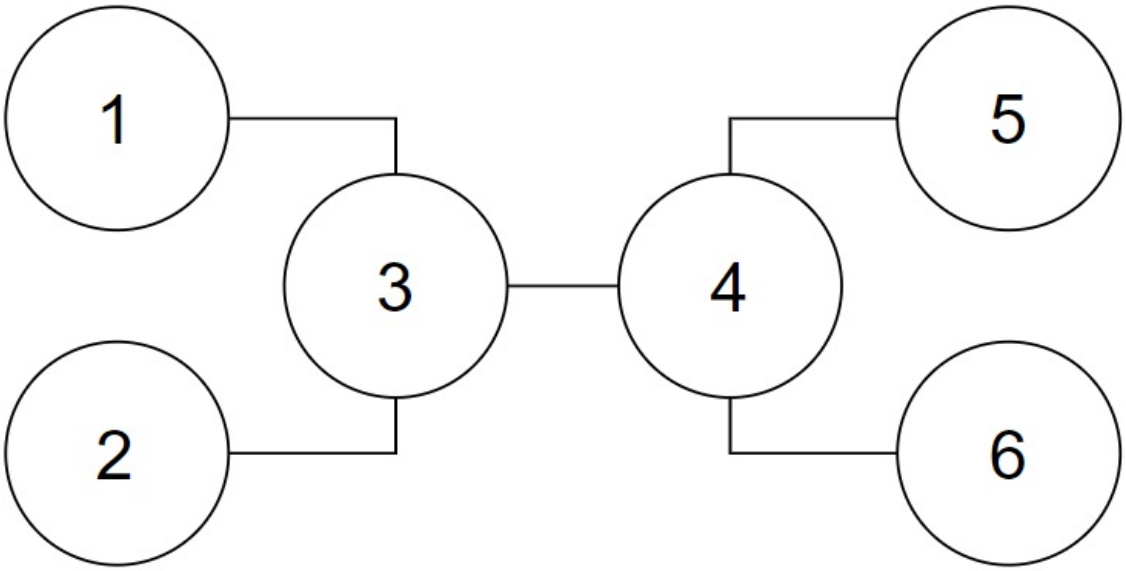
input
6 1 3 2 3 3 4 4 5 4 6
output
-1

Note

This is the picture for the first example. 1, 5, 7 also can be a valid answer.



This is the picture for the second example. You can see that it's impossible to find such root vertex.



E. Product Oriented Recurrence

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

Let $f_x = c^{2x-6} \cdot f_{x-1} \cdot f_{x-2} \cdot f_{x-3}$ for $x \geq 4$.

You have given integers n, f_1, f_2, f_3, and c. Find $f_n \bmod (10^9 + 7)$.

Input

The only line contains five integers n, f_1, f_2, f_3, and c ($4 \leq n \leq 10^{18}$, $1 \leq f_1, f_2, f_3, c \leq 10^9$).

Output

Print $f_n \bmod (10^9 + 7)$.

Examples

input
5 1 2 5 3
output
72900

input
17 97 41 37 11
output
317451037

Note

In the first example, $f_4 = 90$, $f_5 = 72900$.

In the second example, $f_{17} \approx 2.28 \times 10^{29587}$.

F. Maximum Sine

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

You have given integers a, b, p, and q. Let $f(x) = \text{abs}(\sin(\frac{p}{q}\pi x))$.

Find minimum possible integer x that maximizes $f(x)$ where $a \leq x \leq b$.

Input

Each test contains multiple test cases.

The first line contains the number of test cases t ($1 \leq t \leq 100$) — the number of test cases.

The first line of each test case contains four integers a, b, p, and q ($0 \leq a \leq b \leq 10^9$, $1 \leq p, q \leq 10^9$).

Output

Print the minimum possible integer x for each test cases, separated by newline.

Example

input
2 0 3 1 3 17 86 389 995
output
1 55

Note

In the first test case, $f(0) = 0$, $f(1) = f(2) \approx 0.866$, $f(3) = 0$.

In the second test case, $f(55) \approx 0.999969$, which is the largest among all possible values.