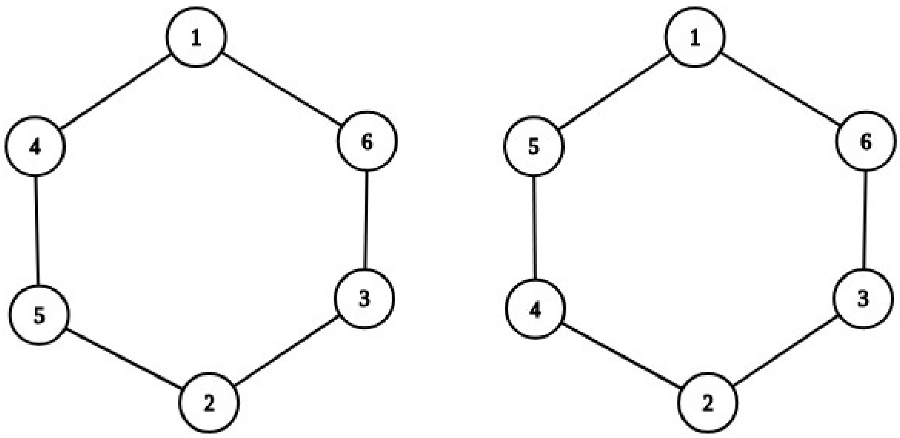
A. Almost Equal

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

You are given integer n. You have to arrange numbers from 1 to $2n$, using each of them exactly once, on the circle, so that the following condition would be satisfied:

For every n consecutive numbers on the circle write their sum on the blackboard. Then any two of written on the blackboard $2n$ numbers differ not more than by 1.

For example, choose $n = 3$. On the left you can see an example of a valid arrangement: $1 + 4 + 5 = 10$, $4 + 5 + 2 = 11$, $5 + 2 + 3 = 10$, $2 + 3 + 6 = 11$, $3 + 6 + 1 = 10$, $6 + 1 + 4 = 11$, any two numbers differ by at most 1. On the right you can see an invalid arrangement: for example, $5 + 1 + 6 = 12$, and $3 + 2 + 4 = 9$, 9 and 12 differ more than by 1.



Input

The first and the only line contain one integer n ($1 \le n \le 10^5$).

Output

If there is no solution, output "NO" in the first line.

If there is a solution, output "YES" in the first line. In the second line output $2n$ numbers — numbers from 1 to $2n$ in the order they will stay in the circle. Each number should appear only once. If there are several solutions, you can output any of them.

Examples

input
```
3
```
output
```
YES
1 4 5 2 3 6
```

input
```
4
```
output
```
NO
```

Note

Example from the statement is shown for the first example.

It can be proved that there is no solution in the second example.

B. Shortest Cycle

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

output: standard output

You are given n integer numbers $a_1, a_2, \dots, a_n$. Consider graph on n nodes, in which nodes i, j ($i \neq j$) are connected if and only if, a_i AND $a_j \neq 0$, where AND denotes the bitwise AND operation.

Find the length of the shortest cycle in this graph or determine that it doesn't have cycles at all.

Input

The first line contains one integer n $(1 \le n \le 10^5)$ — number of numbers.

The second line contains n integer numbers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 10^{18}$).

Output

If the graph doesn't have any cycles, output -1. Else output the length of the shortest cycle.

Examples

input

```
4
3 6 28 9
```

output

```
4
```

input

```
5
5 12 9 16 48
```

output

```
3
```

input

```
4
1 2 4 8
```

output

```
-1
```

Note

In the first example, the shortest cycle is $(9, 3, 6, 28)$.

In the second example, the shortest cycle is $(5, 12, 9)$.

The graph has no cycles in the third example.

C. Palindromic Paths

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

This is an interactive problem

You are given a grid $n \times n$, where n is **odd**. Rows are enumerated from 1 to n from up to down, columns are enumerated from 1 to n from left to right. Cell, standing on the intersection of row x and column y, is denoted by (x, y).

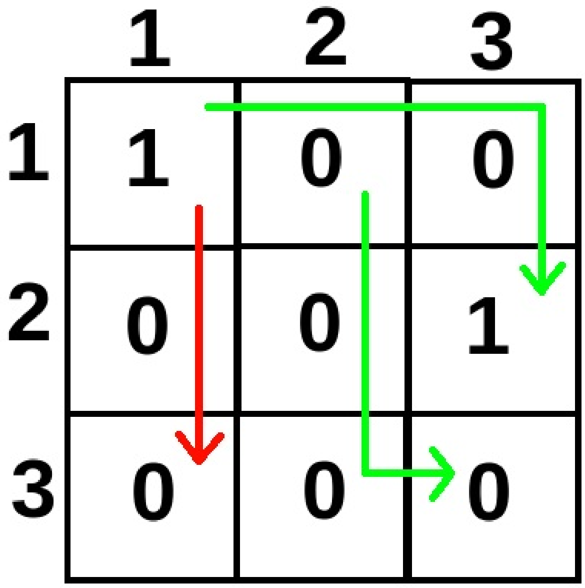
Every cell contains 0 or 1. It is known that the top-left cell contains 1, and the bottom-right cell contains 0.

We want to know numbers in all cells of the grid. To do so we can ask the following questions:

"$?\ x_1\ y_1\ x_2\ y_2$", where $1 \le x_1 \le x_2 \le n$, $1 \le y_1 \le y_2 \le n$, and $x_1 + y_1 + 2 \le x_2 + y_2$. In other words, we output two different cells (x_1, y_1), (x_2, y_2) of the grid such that we can get from the first to the second by moving only to the right and down, and they aren't adjacent.

As a response to such question you will be told if there exists a path between (x_1, y_1) and (x_2, y_2), going only to the right or down, numbers in cells of which form a palindrome.

For example, paths, shown in green, are palindromic, so answer for "$?\ 1\ 1\ 2\ 3$" and "$?\ 1\ 2\ 3\ 3$" would be that there exists such path. However, there is no palindromic path between $(1, 1)$ and $(3, 1)$.



Determine all cells of the grid by asking not more than n^2 questions. It can be shown that the answer always exists.

Input

The first line contains **odd** integer ($3 \le n < 50$) — the side of the grid.

Interaction

You begin the interaction by reading n.

To ask a question about cells $(x_1, y_1), (x_2, y_2)$, in a separate line output "$? \ x_1 \ y_1 \ x_2 \ y_2$".

Numbers in the query have to satisfy $1 \le x_1 \le x_2 \le n$, $1 \le y_1 \le y_2 \le n$, and $x_1 + y_1 + 2 \le x_2 + y_2$. Don't forget to 'flush', to get the answer.

In response, you will receive 1, if there exists a path going from (x_1, y_1) to (x_2, y_2) only to the right or down, numbers in cells of which form a palindrome, and 0 otherwise.

In case your query is invalid or you asked more than n^2 queries, program will print -1 and will finish interaction. You will receive **Wrong answer** verdict. Make sure to exit immediately to avoid getting other verdicts.

When you determine numbers in all cells, output "`!`".

Then output n lines, the i-th of which is a string of length n, corresponding to numbers in the i-th row of the grid.

After printing a query do not forget to output end of line and flush the output. Otherwise, you will 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 documentation for other languages.

Hack Format

To hack, use the following format.

The first line should contain a single odd integer n (side of your grid).

The i-th of n following lines should contain a string of length n corresponding to the i-th row of the grid. Top left element of the grid has to be equal to 1, bottom right has to be equal to 0.

Example

input

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

output

```
? 1 1 1 3
? 1 1 2 3
? 2 1 2 3
? 3 1 3 3
? 2 2 3 3
```

```
? 1 2 3 2
? 1 2 3 3
!
100
001
000
```

D. Almost All

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

You are given a tree with n nodes. You have to write non-negative integers on its edges so that the following condition would be satisfied:

For every two nodes i, j, look at the path between them and count the sum of numbers on the edges of this path. Write all obtained sums on the blackboard. Then every integer from 1 to $\lfloor \frac{2n^2}{9} \rfloor$ has to be written on the blackboard at least once.

It is guaranteed that such an arrangement exists.

Input

The first line contains a single integer n ($1 \leq n \leq 1000$) — the number of nodes.

Each of the next $n - 1$ lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$), meaning that there is an edge between nodes u and v. It is guaranteed that these edges form a tree.

Output

Output $n - 1$ lines, each of form u v x ($0 \leq x \leq 10^6$), which will mean that you wrote number x on the edge between u, v.

Set of edges (u, v) has to coincide with the set of edges of the input graph, but you can output edges in any order. You can also output ends of edges in an order different from the order in input.

Examples

input

```
3
2 3
2 1
```

output

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

input

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

output

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

input

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

output

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

Note

In the first example, distance between nodes 1 and 2 is equal to 2, between nodes 2 and 3 to 1, between 1 and 3 to 3.

In the third example, numbers from 1 to 9 (inclusive) will be written on the blackboard, while we need just from 1 to 5 to pass the test.

E. Expected Value Again

time limit per test: 4 seconds
memory limit per test: 256 megabytes

input: standard input
output: standard output

You are given integers n, k. Let's consider the alphabet consisting of k different elements.

Let **beauty** $f(s)$ of the string s be the number of indexes i, $1 \le i < |s|$, for which prefix of s of length i equals to suffix of s of length i. For example, beauty of the string $abacaba$ equals 2, as for $i = 1, 3$ prefix and suffix of length i are equal.

Consider all words of length n in the given alphabet. Find the expected value of $f(s)^2$ of a uniformly chosen at random word. We can show that it can be expressed as $\frac{P}{Q}$, where P and Q are coprime and Q isn't divided by $10^9 + 7$. Output $P \cdot Q^{-1} \bmod 10^9 + 7$.

Input

The first and the only line contains two integers n, k ($1 \le n \le 10^5$, $1 \le k \le 10^9$) — the length of a string and the size of alphabet respectively.

Output

Output a single integer — $P \times Q^{-1} \bmod 10^9 + 7$.

Examples

input
2 3
output
333333336

input
1 5
output
0

input
100 1
output
9801

input
10 10
output
412377396

Note

In the first example, there are 9 words of length 2 in alphabet of size 3 — aa, ab, ac, ba, bb, bc, ca, cb, cc. 3 of them have beauty 1 and 6 of them have beauty 0, so the average value is $\frac{1}{3}$.

In the third example, there is only one such word, and it has beauty 99, so the average value is 99^2.

F. Beauty of a Permutation

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

Define the beauty of a permutation of numbers from 1 to n $(p_1, p_2, \ldots, p_n)$ as number of pairs (L, R) such that $1 \le L \le R \le n$ and numbers $p_L, p_{L+1}, \ldots, p_R$ are consecutive $R - L + 1$ numbers in some order. For example, the beauty of the permutation $(1, 2, 5, 3, 4)$ equals 9, and segments, corresponding to pairs, are $[1]$, $[2]$, $[5]$, $[4]$, $[3]$, $[1, 2]$, $[3, 4]$, $[5, 3, 4]$, $[1, 2, 5, 3, 4]$.

Answer q independent queries. In each query, you will be given integers n and k. Determine if there exists a permutation of numbers from 1 to n with beauty equal to k, and if there exists, output one of them.

Input

The first line contains a single integer q ($1 \le q \le 10\,000$) — the number of queries.

Follow q lines. Each line contains two integers n, k ($1 \le n \le 100$, $1 \le k \le \frac{n(n+1)}{2}$) — the length of permutation and needed beauty respectively.

Output

For a query output "NO", if such a permutation doesn't exist. Otherwise, output "YES", and in the next line output n numbers — elements of permutation in the right order.

Examples

input

```
4
1 1
5 6
5 8
5 10
```

output

```
YES
1
YES
2 4 1 5 3
NO
YES
2 3 1 4 5
```

input

```
2
4 10
100 1
```

output

```
YES
1 2 3 4
NO
```

Note

Let's look at the first example.

The first query: in (1) there is only one segment consisting of consecutive numbers — the entire permutation.

The second query: in $(2, 4, 1, 5, 3)$ there are 6 such segments: $[2]$, $[4]$, $[1]$, $[5]$, $[3]$, $[2, 4, 1, 5, 3]$.

There is no such permutation for the second query.

The fourth query: in $(2, 3, 1, 4, 5)$ there are 10 such segments: $[2]$, $[3]$, $[1]$, $[4]$, $[5]$, $[2, 3]$, $[2, 3, 1]$, $[2, 3, 1, 4]$, $[4, 5]$, $[2, 3, 1, 4, 5]$.

Codeforces (c) Copyright 2010-2022 Mike Mirzayanov
The only programming contests Web 2.0 platform