A. Reorder the Array

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

You are given an array of integers. Vasya can permute (change order) its integers. He wants to do it so that as many as possible integers will become on a place where a smaller integer used to stand. Help Vasya find the maximal number of such integers.

For instance, if we are given an array $[10, 20, 30, 40]$, we can permute it so that it becomes $[20, 40, 10, 30]$. Then on the first and the second positions the integers became larger ($20 > 10$, $40 > 20$) and did not on the third and the fourth, so for this permutation, the number that Vasya wants to maximize equals 2. Read the note for the first example, there is one more demonstrative test case.

Help Vasya to permute integers in such way that the number of positions in a new array, where integers are greater than in the original one, is maximal.

Input

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

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

Output

Print a single integer — the maximal number of the array's elements which after a permutation will stand on the position where a smaller element stood in the initial array.

Examples

input

```
7
10 1 1 1 5 5 3
```

output

```
4
```

input

```
5
1 1 1 1 1
```

output

```
0
```

Note

In the first sample, one of the best permutations is $[1, 5, 5, 3, 10, 1, 1]$. On the positions from second to fifth the elements became larger, so the answer for this permutation is 4.

In the second sample, there is no way to increase any element with a permutation, so the answer is 0.

B. Pave the Parallelepiped

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

You are given a rectangular parallelepiped with sides of positive integer lengths A, B and C.

Find the number of different groups of three integers (a, b, c) such that $1 \leq a \leq b \leq c$ and parallelepiped $A \times B \times C$ can be paved with parallelepipeds $a \times b \times c$. Note, that all small parallelepipeds **have to be rotated in the same direction**.

For example, parallelepiped $1 \times 5 \times 6$ can be divided into parallelepipeds $1 \times 3 \times 5$, but can not be divided into parallelepipeds $1 \times 2 \times 3$.

Input

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

Each of the next t lines contains three integers A, B and C ($1 \leq A, B, C \leq 10^5$) — the sizes of the parallelepiped.

Output

For each test case, print the number of different groups of three points that satisfy all given conditions.

Example

input

```
4
1 1 1
1 6 1
2 2 2
100 100 100
```

output

```
1
4
4
165
```

Note

In the first test case, rectangular parallelepiped $(1,1,1)$ can be only divided into rectangular parallelepiped with sizes $(1,1,1)$.

In the second test case, rectangular parallelepiped $(1,6,1)$ can be divided into rectangular parallelepipeds with sizes $(1,1,1)$, $(1,1,2)$, $(1,1,3)$ and $(1,1,6)$.

In the third test case, rectangular parallelepiped $(2,2,2)$ can be divided into rectangular parallelepipeds with sizes $(1,1,1)$, $(1,1,2)$, $(1,2,2)$ and $(2,2,2)$.

C. Guess two numbers

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

This is an interactive problem.

Vasya and Vitya play a game. Vasya thought of two integers a and b from 1 to n and Vitya tries to guess them. Each round he tells Vasya two numbers x and y from 1 to n. If both $x = a$ and $y = b$ then Vitya wins. Else Vasya must say one of the three phrases:

1. x is less than a;
2. y is less than b;
3. x is greater than a or y is greater than b.

Vasya can't lie, but if multiple phrases are true, he may choose any of them. For example, if Vasya thought of numbers 2 and 4, then he answers with the phrase 3 to a query $(3,4)$, and he can answer with the phrase 1 or phrase 3 to a query $(1,5)$.

Help Vitya win in no more than 600 rounds.

Input

The first line contains a single integer n ($1 \le n \le 10^{18}$) — the upper limit of the numbers.

Interaction

First, you need to read the number n, after that you can make queries.

To make a query, print two integers: x and y ($1 \le x, y \le n$), then `flush` the output.

After each query, read a single integer ans ($0 \le ans \le 3$).

If $ans > 0$, then it is the number of the phrase said by Vasya.

If $ans = 0$, it means that you win and your program should terminate.

If you make more than 600 queries or make an incorrect query, you will get `Wrong Answer`.

Your solution will get `Idleness Limit Exceeded`, if you don't print anything or forget to `flush` the output.

To `flush` you need to do the following right after printing a query and a line end:

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal;
- For other languages see documentation.

Hacks format

For hacks, use the following format:

In the first line, print a single integer n ($1 \le n \le 10^{18}$) — the upper limit of the numbers.

In the second line, print two integers a and b ($1 \le a, b \le n$) — the numbers which Vasya thought of.

In the third line, print a single integer m ($1 \le m \le 10^5$) — the number of instructions for the interactor.

In each of the next m lines, print five integers: x_i, y_i, r_i^{12}, r_i^{13}, and r_i^{23} ($1 \le x_i, y_i \le n$), where r_i^{ST} equals to either number S or number T.

While answering the query x y, the interactor finds a number i from 1 to n with the minimal value $|x - x_i| + |y - y_i|$. If multiple numbers can be chosen, the least i is preferred. Then the interactor answers to the query, but if there are two phrases S and T that can be given, then r_i^{ST} is chosen.

For example, the sample test data file contains the following:

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

Example

input

```
5
3
3
2
1
0
```

output

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

Note

Let's analyze the sample test. The chosen numbers are 2 and 4. The interactor was given two instructions.

For the query $(4, 3)$, it can return 2 or 3. Out of the two instructions the second one is chosen, so the interactor returns $a_2^{23} = 3$.

For the query $(3, 4)$, it can return only 3.

For the query $(3, 3)$, it can return 2 or 3. Out of the two instructions the first one is chosen (since in case of equal values, the least number is preferred), so the interactor returns $a_1^{23} = 2$.

For the query $(1, 5)$, it can return 1 or 3. Out of the two instructions the first one is chosen, so the interactor returns $a_1^{13} = 1$.

In the fifth query $(2, 4)$, the numbers are guessed correctly, the player wins.

D. Ants

time limit per test: 3 seconds

memory limit per test: 768 megabytes

input: standard input

output: standard output

There is a tree with n vertices. There are also m ants living on it. Each ant has its own color. The i-th ant has two favorite pairs of vertices: (a_i, b_i) and (c_i, d_i). You need to tell if it is possible to paint the edges of the tree in m colors so that every ant will be able to walk between vertices from one of its favorite pairs using only edges of his color; if it is possible, you need to print which pair every ant should use.

Input

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

Each of the next $n - 1$ lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$), meaning that there is an edge between vertices u_i and v_i.

The next line contains a single integer m ($1 \le m \le 10^4$) — the number of ants.

Each of the next m lines contains four integers a_i, b_i, c_i, and d_i ($1 \le a_i, b_i, c_i, d_i \le n$, $a_i \ne b_i, c_i \ne d_i$), meaning that pairs (a_i, b_i) and (c_i, d_i) are favorite for the i-th ant.

Output

Print "NO" (without quotes) if the wanted painting is impossible.

Otherwise, print "YES" (without quotes). Print m lines. On the i-th line, print 1 if the i-th ant will use the first pair and 2 otherwise. If there are multiple answers, print any.

Examples

input

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

output

```
YES
2
1
2
```

input

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

output

```
NO
```

Note

In the sample, the second and the third edge should be painted in the first color, the first and the fifth should be painted in the second color, and the fourth should be painted in the third color.

E. Mini Metro

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

In a simplified version of a "Mini Metro" game, there is only one subway line, and all the trains go in the same direction. There are n stations on the line, a_i people are waiting for the train at the i-th station at the beginning of the game. The game starts at the beginning of the 0-th hour. At the end of each hour (couple minutes before the end of the hour), b_i people instantly arrive to the i-th station. If at some moment, the number of people at the i-th station is larger than c_i, you lose.

A player has several trains which he can appoint to some hours. The capacity of each train is k passengers. In the middle of the appointed hour, the train goes from the 1-st to the n-th station, taking as many people at each station as it can accommodate. A train can not take people from the i-th station if there are people at the $i-1$-th station.

If multiple trains are appointed to the same hour, their capacities are being added up and they are moving together.

The player wants to stay in the game for t hours. Determine the minimum number of trains he will need for it.

Input

The first line contains three integers n, t, and k ($1 \le n, t \le 200, 1 \le k \le 10^9$) — the number of stations on the line, hours we want to survive, and capacity of each train respectively.

Each of the next n lines contains three integers a_i, b_i, and c_i ($0 \le a_i, b_i \le c_i \le 10^9$) — number of people at the i-th station in the beginning of the game, number of people arriving to i-th station in the end of each hour and maximum number of people at the i-th station allowed respectively.

Output

Output a single integer number — the answer to the problem.

Examples

input

```
3 3 10
2 4 10
3 3 9
4 2 8
```

output

```
2
```

input

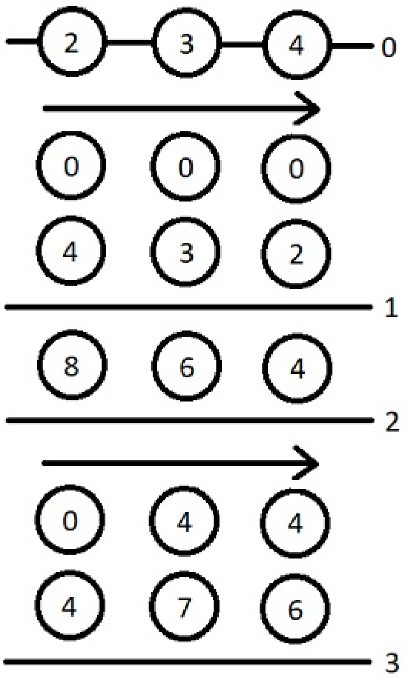
```
4 10 5
1 1 1
```

```
1 0 1
0 5 8
2 7 100
```

output

```
12
```

Note



Let's look at the sample. There are three stations, on the first, there are initially 2 people, 3 people on the second, and 4 people on the third. Maximal capacities of the stations are 10, 9, and 8 respectively.

One of the winning strategies is to appoint two trains to the first and the third hours. Then on the first hour, the train takes all of the people from the stations, and at the end of the hour, 4 people arrive at the first station, 3 on the second, and 2 on the third.

In the second hour there are no trains appointed, and at the end of it, the same amount of people are arriving again.

In the third hour, the train first takes 8 people from the first station, and when it arrives at the second station, it takes only 2 people because it can accommodate no more than 10 people. Then it passes by the third station because it is already full. After it, people arrive at the stations once more, and the game ends.

As there was no such moment when the number of people at a station exceeded maximal capacity, we won using two trains.

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