

## Problem A. The Lawyer

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Byteasar is an advocate, co-owner of the law firm *Byteasar and Associates*. He is one of the most sought-after members of the Byteotian Bar. Not surprisingly, he is always extremely busy. Every day he is involved in a number of meetings and has long since ceased to control whether he will be able to participate in all of them. Therefore he has hired a secretary, whose job is to help him to control this chaos. Byteasar decided that every day he will be taking part in two meetings only, however his participation would be complete, that is, from the very beginning to the very end. The remaining meetings will be taken care of by assistants, who are quite plentiful at Byteasar's office.

Unfortunately, it is sometimes difficult even to find two meetings which do not overlap in Byteasar's busy schedule. We assume that two meetings do not overlap, if one of them starts *strictly after* the previous has finished. Help Byteasar's secretary and write a program that can deal with this problem.

### Input

The first line of input contains two integers  $n$  and  $m$  ( $2 \leq n \leq 500\,000$ ,  $1 \leq m \leq 20$ ) describing the number of meetings in Byteasar's schedule and the number of days included in it.

Each of the next  $n$  lines describes one meeting. Description of the meeting consists of three integers  $a_i, b_i, d_i$  ( $1 \leq a_i < b_i \leq 80\,000\,000$ ,  $1 \leq d_i \leq m$ ) which indicate that on the day  $d_i$  Byteasar has a meeting scheduled that starts exactly  $a_i$  milliseconds after midnight and ends  $b_i$  milliseconds after midnight.

### Output

Your program should output  $m$  lines. The  $i$ -th of these lines should contain information, whether Byteasar is able to attend two meetings on the  $i$ -th day. In case it is not possible, one word "NIE" (i.e., Polish for "no") should be produced. Otherwise, the word "TAK" (i.e., Polish for "yes") should be output, followed by the numbers of the two meetings in which Byteasar can participate. Meetings are numbered from 1 to  $n$ , in accordance to their order at the input. The first of these two meetings should start earlier. The second meeting should start at least a millisecond after the first one is finished.

If there are multiple correct answers, your program should output any one of them.

### Examples

standard input	standard output
6 3	TAK 1 6
3 5 1	NIE
2 4 2	NIE
1 8 1	
6 7 3	
3 5 2	
7 12 1	

## Problem B. Petrol

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            1.8 seconds  
Memory limit:         256 mebibytes

Byteasar is employed in the logistics department of Byteonian petroleum giant Byteoil. His job is to plan fuel deliveries to petrol stations.

There are  $n$  intersections in Byteotia (marked by numbers from 1 to  $n$ ) and  $m$  two-way roads connecting certain pairs of intersections. There are Byteoil petrol stations located at some intersections.

Byteoil transport fleet consists of petrol tankers with various capacities of fuel tanks. Each tanker consumes 1 litre of petrol per kilometre of distance travelled. It can therefore be assumed that a tanker having tank capacity of  $b$  litres can cover a maximum distance of  $b$  kilometres without the need to refuel. Drivers cannot use the fuel carried by the tanker in the main tank, however they can refuel at any Byteoil petrol stations free of charge.

Byteasar's work consists of repeatedly answering the following queries: Is a petrol tanker with a fuel tank of capacity of  $b$  litres capable of covering the distance of from a petrol station located at the intersection  $x$  to a petrol station located at the intersection  $y$ ? Tanker with a fuel tank of capacity of  $b$  litres cannot cover a distance of more than  $b$  kilometres, during which there will be no Byteoil petrol station. Tankers starting point is always located at an intersection where Byteoil petrol station is present, and also all the trips end at an intersection with Byteoil petrol station.

Help Byteasar to provide an automated reply to his logistic queries.

### Input

The first line of input contains three integers  $n$ ,  $s$  and  $m$  ( $2 \leq s \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$ ), denoting the number of intersections, the number of petrol stations and the number of roads in Byteotia. The second line contains a sequence of  $s$  pairwise distinct integers  $c_1, c_2, \dots, c_s$  ( $1 \leq c_i \leq n$ ), denoting the intersections where Byteoil stations are located.

The next  $m$  lines describe the roads in Byteotia;  $i$ -th of these lines contains three integers  $u_i$ ,  $v_i$  and  $d_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $1 \leq d_i \leq 10\,000$ ), indicating that the  $i$ -th road has a length of  $d_i$  kilometres and connects the intersection  $u_i$  with the intersection  $v_i$ . Each pair of intersections is connected by at most one road.

The next line contains one integer  $q$  ( $1 \leq q \leq 200\,000$ ), denoting the number of queries. The consecutive  $q$  lines hold the descriptions of queries;  $i$ -th of these lines contains three integers  $x_i$ ,  $y_i$  and  $b_i$  ( $1 \leq x_i, y_i \leq n$ ,  $x_i \neq y_i$ ,  $1 \leq b_i \leq 2 \cdot 10^9$ ) indicating query concerning the possibility of a tanker with a capacity of  $b_i$  litres to cover the distance from a petrol station at the intersection  $x_i$  to the station at the intersection of  $y_i$ . It can be assumed that at both intersections  $x_i$ ,  $y_i$  Byteoil petrol stations are located.

### Output

Your program should output exactly  $q$  lines. The  $i$ -th of these lines should contain one word "TAK" (i.e., *yes*) or "NIE" (i.e., *no*), depending on whether the tanker with a fuel tank with a capacity of  $b_i$  is able to travel from the intersection  $x_i$  to the intersection  $y_i$ .

## Examples

standard input	standard output
6 4 5	TAK
1 5 2 6	TAK
1 3 1	TAK
2 3 2	NIE
3 4 3	
4 5 5	
6 4 5	
4	
1 2 4	
2 6 9	
1 5 9	
6 5 8	

## Problem C. The Prices

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            2.5 seconds  
Memory limit:         256 mebibytes

Byteasar works as a Purchasing Manager at one of Byteotian restaurants. Every evening he receives a shopping list from his manager. Food products must be purchased the next day, in the morning. Byteasar should buy exactly one piece of each product from the list. Manager is always pressing that the total costs are as little as possible.

Byteasar sits down in the evening with his computer and checks the prices of all needed products at local grocery wholesalers. He also knows the cost of the trips from the restaurant to each wholesale and back. Now Byteasar must decide which products are to be purchased at each of the warehouses.

For each wholesale, where Byteasar decided to buy some products, he does as follows. He would go from the restaurant to the warehouse, will do the shopping and immediately bring the purchased products back to the restaurant. Luckily, the boot of his car is big enough that it eliminates the need to visit any of the warehouses more than once, as all the purchased goods can be delivered in one go. Food products are highly perishable, so during one trip Byteasar can make purchases only at one warehouse.

Write a program that will help Byteasar to calculate the cheapest way of making all the purchases.

### Input

The first line of input contains two integers  $n$  and  $m$  ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 16$ ) denoting the number of wholesalers and the number of products that Byteasar should buy. Next  $n$  lines contain descriptions of the prices at individual wholesalers.

The first number in the  $i$ -th of these lines,  $d_i$  ( $1 \leq d_i \leq 1\,000\,000$ ), describes the trip cost from the restaurant to the  $i$ -th warehouse (including the return cost). It is followed by a sequence of  $m$  integers  $c_{i,1}, c_{i,2}, \dots, c_{i,m}$  ( $1 \leq c_{i,j} \leq 1\,000\,000$ ): number  $c_{i,j}$  denotes the price of  $j$ -th product in the  $i$ -th warehouse.

### Output

Your program should output a single line containing a single integer denoting the sum of the product prices and the cost of the trips to warehouses selected by Byteasar in the cheapest possible purchase plan.

### Examples

standard input	standard output
3 4	16
5 7 3 7 9	
2 1 20 3 2	
8 1 20 1 1	

### Note

Byteasar buys the product number 2 in the first warehouse, and all the other products in the second one. So he does not have to visit the third warehouse.

## Problem D. Divisors

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3.5 seconds  
Memory limit: 256 mebibytes

You are given a sequence of  $n$  integers  $a_1, a_2, \dots, a_n$ . You should determine the number of such ordered pairs  $(i, j)$ , that  $i, j \in \{1, \dots, n\}$ ,  $i \neq j$  and  $a_i$  is a divisor of  $a_j$ .

### Input

The first line of input contains one integer  $n$  ( $1 \leq n \leq 2\,000\,000$ ). The second line contains a sequence of  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 2\,000\,000$ ).

### Output

In the first and only line of output should contain one integer, denoting the number of pairs sought.

### Examples

standard input	standard output
5 2 4 5 2 6	6

### Note

There are 6 pairs with the specified properties:  $(1, 2)$ ,  $(1, 4)$ ,  $(1, 5)$ ,  $(4, 1)$ ,  $(4, 2)$ ,  $(4, 5)$ .

## Problem E. Euclidean Nim

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Euclid and Pythagoras are pseudonyms of two Byteotians for their love of mathematical puzzles. Lately, they spend evenings playing the following game. There is a stack of  $n$  stones on the table. Friends perform alternating moves. Euclid's move consists of taking any positive multiple of  $p$  stones from the stack (providing the stack contains at least  $p$  stones) or adding exactly  $p$  stones to the stack—adding the stones is possible, however, only in case the stack contains less than  $p$  stones. Pythagoras' move is analogical, except that either he takes a multiple of  $q$  stones, or adds exactly  $q$  stones. The winner is the player who empties the stack. Euclid begins the game.

Friends wonder whether they have worked out this game perfectly. Help them and write a program that will state what should be the result of the game, providing both players are making optimal moves.

### Input

The first line of input contains one integer  $t$  ( $1 \leq t \leq 1000$ ) denoting the number of test cases described in the following part of the input. Description of one test case consists of one line containing three integers  $p$ ,  $q$  and  $n$  ( $1 \leq p, q, n \leq 10^9$ ).

### Output

Output should include exactly  $t$  lines containing answers to the subsequent test cases from input. The answer should be one letter 'E' (if Euclid could bring about his victory, regardless of the Pythagoras' movements), 'P' (if Pythagoras could bring about his victory, regardless of Euclid's moves) or 'R' (for *remis*, i.e. *draw* in Polish, if the game will be played infinitely).

### Examples

standard input	standard output
4	P
3 2 1	P
2 3 1	E
3 4 5	R
2 4 3	

### Note

In the game from the first test case Euclid must add three stones to the stack in his move. Thanks to that Pythagoras can collect all 4 stones in his move and thus win.

## Problem G. Global Warming

Input file: *standard input*  
Output file: *standard output*  
Time limit: 1 second  
Memory limit: 256 mebibytes

Professor Byteasar prepares a report for Interbyteotian Commission for Climate Change. This report should demonstrate clearly, unambiguously and without any doubt, the impact of Byteotian population on climate changes in the region. Although the Professor has a lot of empirical data, but to pervade the mainstream media, it not enough to present substantive arguments. Equally important is to present data clearly and correctly targeted. To this end, he wants to select the data, to be presented on the main chart in the report, thoughtfully and in a deliberate way.

The key graph will contain information about the average air temperature throughout the years. The Professor has data available concerning the mean annual temperatures for the last  $n$  years. He wants to describe this graph using the following comment: “in the year  $r_{min}$  the temperature was the lowest, and in the year  $r_{max}$  it was the highest, therefore, it is clear that ...”. Unfortunately, he fears that the same minimum or maximum temperature can occur several times throughout this period, and therefore such information could undermine the clearness of his statement.

Professor has therefore decided to present only part of the data on his chart. His idea is to select the year range in such a way that this interval will include exactly one year with a minimum temperature *within that range* and exactly one year with a maximum temperature also *within that range*. The selected range may not include the year with the globally maximum or minimum average temperature in the last  $n$  years (or none of them). Of course, the Professor would like to put as much data as possible on this chart, so he is interested in the longest year span.

### Input

The first line of input contains one integer  $n$  ( $1 \leq n \leq 500\,000$ ), denoting the number of years for which the Professor knows the average temperatures. The second line contains a sequence of  $n$  integers  $t_1, t_2, \dots, t_n$  ( $-10^9 \leq t_i \leq 10^9$ ). The number  $t_i$  denotes the average temperature in the  $i$ -th year.

### Output

The output should contain two integers  $l$  and  $k$ . They indicate that the longest interval satisfying the Professor’s conditions is of length of  $l$  years, and the earliest year in which such an interval may begin is  $k$ .

### Example

standard input	standard output
10 8 3 2 5 2 3 4 6 3 6	6 4

### Note

The chart will present temperatures 5, 2, 3, 4, 6, 3. This interval contains exactly one year with a minimum temperature of 2 and one year with a maximum temperature of 6.

## Problem J. The Cave

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            2.6 seconds  
Memory limit:         256 mebibytes

A group of speleologists plans to explore a recently discovered cave complex. The cave complex consists of  $n$  chambers numbered from 1 to  $n$ . The chambers are connected by means of  $n - 1$  corridors in such a way that any chamber can be reached from any other. Each corridor connects exactly two chambers.

The cave will be explored by a group of  $m$  speleologists. For simplicity we number them by integers from 1 to  $m$ . Each speleologist presented his requirements concerning the area of the cave which he would like to explore. Speleologist  $i$  would like to begin his exploration in the chamber  $a_i$ , finish exploring in the chamber  $b_i$  and traverse a maximum of  $d_i$  corridors on his way (passing the same corridor each time is to be counted separately). Byteasar, head of the expedition, would like all the researchers to be able to meet at a certain point in time in order to exchange their observations. For this reason, he is wondering whether he could choose one of the chambers of the cave system, and plan the routes of all the speleologists in such a way, that they all pass through the selected chamber. Of course planned routes must meet the requirements described initially by the researchers.

### Input

The first line of input contains one integer  $t$  ( $1 \leq t \leq 1000$ ) that specifies the number of test cases. It is followed by the descriptions of the individual cases. Description of a single case begins with a line including two integers  $n$  and  $m$  ( $2 \leq n, m \leq 300\,000$ ), which describe the number of chambers in the cave system and the number of speleologists, respectively. The next  $n - 1$  lines describe the cave corridors. Each of them contains two integers  $u_i, w_i$  ( $1 \leq u_i, w_i \leq n$ ), which indicate that the chambers  $u_i$  and  $w_i$  are connected by a direct corridor.

Next  $m$  lines describe the speleologists' requirements. The  $i$ -th of these lines contains three integers  $a_i, b_i, d_i$  ( $1 \leq a_i, b_i \leq n, 1 \leq d_i \leq 600\,000$ ). They indicate that the speleologist  $i$  will begin by starting to explore chamber  $a_i$ , finish in chamber  $b_i$ , and moving between chambers passes through at most  $d_i$  corridors. You may always assume that it is possible to move from chamber  $a_i$  to  $b_i$  traversing not more than  $d_i$  corridors. Both the sum of values  $n$  for all the test cases, as well as the total value of  $m$  does not exceed 300 000.

### Output

Your program should output exactly  $t$  lines. The  $i$ -th line should contain the answer to the  $i$ -th test case from input. In case it is possible to set speleologists' routes in such a manner, so that they all run through one common chamber, one word TAK (i.e., Polish for *yes*) should be produced, followed by the number of the chamber where the meeting is to take place. Otherwise, your program should output only one word: "NIE". If there are multiple correct answers, your program should output any of them.

## Examples

standard input	standard output
2	TAK 2
5 3	NIE
1 2	
2 3	
2 4	
3 5	
1 4 2	
5 5 5	
3 2 1	
3 2	
1 2	
2 3	
1 1 2	
3 3 1	

## Problem K. The Captain

Input file:            *standard input*  
Output file:           *standard output*  
Time limit:            2.5 seconds  
Memory limit:         256 mebibytes

Captain Byteasar compasses the waters of Byteic Sea together with his irreplaceable First Officer Bytec. There are  $n$  islands in the Byteic Sea, which we numbered from 1 to  $n$ . Captain's ship has docked at the island number 1. Captain's expedition plan is to sail to the island number  $n$ .

During the voyage, the ship always moves in one of four directions of the world: north, south, east or west. At any time it is either the Captain or the First Officer standing at the helm. Every time the ship will perform  $90^\circ$  turn, they would change at the helm.

Along its way, the vessel may stop at other islands. After each stop, the Captain can decide whether he takes control of the helm first, or not. In other words, for each route leg, leading from an island to another one, one of the sailors stands at the helm while the ship is travelling north or south, and the other controls it while it is moving east or west. In particular, if a given fragment of the route runs exactly in one of the four directions of the world, the ship is controlled by only one of the sailors.

The captain is now considering how to plan a route of the forthcoming voyage and how to divide labour in such a way to spend as little time at the helm. At the same time the Captain does not care how long the calculated route would be. It is assumed that the vessel is sailing at a constant rate of one unit per hour.

### Input

The first line of input contains a single integer  $n$  ( $2 \leq n \leq 200\,000$ ) denoting the number of islands in the sea. For simplicity, we introduce a coordinate system onto Byteic Sea with axes parallel to the directions of the world. Every island is represented as a single point. Subsequent  $n$  lines contain descriptions of the islands:  $i$ -th line contains two integers  $x_i, y_i$  ( $0 \leq x_i, y_i \leq 1\,000\,000\,000$ ) denoting the coordinates of  $i$ -th island in the sea. Each island bears different coordinates.

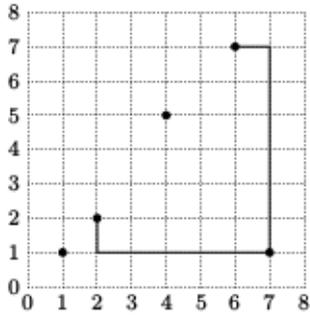
### Output

Your program should output a single integer denoting the least number of hours the Captain will have to steer the ship on the route from the island number 1 to the number  $n$ .

### Examples

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

## Note



The Captain may designate a route that is indicated in the figure. During the voyage from island 1 (coordinates: (2,2)) to the island 4 (coordinates: (7,1)) the Captain controls the ship only for an hour, while the ship is sailing south. During the second leg of the trip the Captain controls the vessel only when it is moving east.

## Problem L. Astroturfing

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Well-known Internet company «Dalstolb» created an on-line marketplace, where individuals and companies can sell some products. After purchasing and testing the product, buyer can rate the seller with an integer from 1 to 10. For each seller, website displays number of ratings  $N$  and their average  $A$  so that prospective buyers can choose better sellers.

If a company is not satisfied with its rating, it might either improve the quality of their goods and services, or create some fake accounts and give high ratings from those accounts to itself. That practice is called *astroturfing*.

«Dalstolb» is aware of astroturfing and implemented following limits to make it harder: each buyer may give only one rating to a given seller; all ratings strictly greater than twice the current average are flagged for manual inspection.

Your program must calculate the minimum number of fake account required to raise the rating to at least  $B$ , while not triggering manual inspection.

Note that all calculations by «Dalstolb» server are assumed to be absolutely exact, with no loss of precision at all.

In the first example, ratings and averages are as follows:

Fake account no.	Fake rating	Average rating
1	4	2.40000
2	4	2.66667
3	5	3.00000
4	6	3.37500
5	6	3.66667
6	7	4.00000

### Input

Input file contains integers  $N$ ,  $A$ ,  $B$  ( $1 \leq N \leq 10^8$ ,  $1 \leq A < B \leq 9$ ).

### Output

Output file must contain a single integer — minimum number of fake accounts.

### Examples

standard input	standard output
4 2 4	6
182736 1 9	8040384
2 8 9	2

## Problem M. Error Detection

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2.6 seconds  
Memory limit: 256 mebibytes

FEFU computer scientists use a duplication method to detect errors in network data transfer. Each bit of data is duplicated. The duplicate follows the original bit. Each byte therefore is transferred as a message of two bytes. The first byte of the message encodes high bits of the original byte, the second byte encodes low bits.

For example, a byte 67 is encoded as follows:

1.  $67_{10} = 01000011_2$ ;
2. the sequence 01000011 is encoded as 0011000000001111;
3. the sequence 0011000000001111 is split into 00110000 and 00001111;
4.  $00110000_2 = 48_{10}$ ,  $00001111_2 = 15_{10}$ , so the result is 4815

Scientists received two bytes. Now they need to find out what the original byte was.

### Input

Input file contains two integers designating two received bytes. Both input integers are in the range from 0 to 255.

### Output

Output file must contain a single integer — the original byte, or  $-1$  if there was an error during transfer.

### Examples

standard input	standard output
48 15	67
48 7	-1

## Problem N. Hacktris

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Vladivostok Hackathon was an event where several student teams were invited to invent and prototype their fresh and useful software ideas in several days.

Young programmer Vasya and his team suggested original and exciting game: Tetris. Game implementation turned to be more complex than young programmers expected, so they simplified the rules.

A game is played on a rectangular field of  $H$  rows and  $W$  columns of cells. Each cell is either occupied or free.

A piece takes a single cell. New piece appears at some cell on the top row, and starts to fall down in steps. On every step a player can choose any free cell of the three cells in a row below the piece — directly under it, a single cell to the right or a single cell to the left.

A piece stops dropping when it either reaches the bottom row, or the row below it is occupied and the player did not choose to move the piece right or left. After the piece stops, next piece immediately appears. Nothing else happens in the game — no rows disappearing, no score, etc.

While Vasya's team was presenting the game to the jury, it crashed and refused to start again. The only proof that the game ever worked is the log file which it wrote on the previous run. Log file contains data about  $N$  pieces — the column where the piece appeared and the column where it stopped.

Your program must, given field a description and a game log, determine whether the log could be the result of a correct game, and, if not, what is the first piece which could not fall as described in the log.

In the sample 2, after first 6 pieces fall and 7th one appears, the field looks as follows:

```
.#. . . .  
..#. . . .  
..#. . . .  
..#. . . .  
.##. . #
```

It is impossible to move the piece to the 6th column, so the log is incorrect.

### Input

Input file contains integers  $WHN$ , followed by  $N$  pairs of integers  $x_1x_2$  — initial and final column of each piece ( $1 \leq W, H \leq 10^4$ ,  $1 \leq N \leq 10^6$ ,  $1 \leq x_1, x_2 \leq W$ ).

### Output

Output file must contain a single integer — the number of first incorrectly positioned piece, or 0 if all pieces are positioned correctly.

## Examples

standard input	standard output
1 10 1 1 1	0
6 5 7 1 3 5 2 6 3 3 3 2 6 4 3 2 6	7