# 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 $c_1, c_2, \ldots, c_s$ ($1 \leq c_i \leq n$), 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 $c_{i,1}, c_{i,2}, \ldots, c_{i,m}$ ($1 \leq c_{i,j} \leq 1\,000\,000$): 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<br>5 7 3 7 9<br>2 1 20 3 2<br>8 1 20 1 1 | 16 |

## 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 $a_1, a_2, \ldots, a_n$. You should determine the number of such ordered pairs $(i, j)$, that $i, j \in \{1, \ldots, n\}$, $i \neq j$ 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 $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 2\,000\,000$).

## 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<br>2 4 5 2 6 | 6 |

## Note

There are 6 pars 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 $t$ ($1 \le t \le 1000$) 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 $n$ ($1 \le p, q, n \le 10^9$).

## 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 F. Pillars

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

Byteasar holds the administrating manager position at a large warehouse. Anticipating a severe winter, he decided to install underfloor heating system in the warehouse.

Warehouse plan is a rectangle of even dimensions $n \times m$ divided into individual square units. Most of the individual squares comprise warehouse space, but some of it is occupied by massive pillars providing additional constructional support for the warehouse structure. Each pillar occupies a square $2 \times 2$ on the warehouse plan, composed of individual square units. Pillars are not arranged too densely—is known that any two of them are separated by at least 6 units (in Euclidean metric). Additionally, the centre of each pillar is located at least by 3 units away from each of the outer warehouse walls.

Heating will be accomplished by using one heating pipe installed under the floor of the warehouse. The pipe is to run through the centres of all individual square units except the square units occupied by pillars. Each pipe section must run parallel to one of the walls of the hall and the turns could be located only at the centres of the square units. The pipe must begin and end in the same place. At this point the cold water would be discharged outside and hot water fed into the pipe.

Byteasar has asked you to plan the pipe trajectory in the warehouse. To help you, he introduced rectangular coordinate system onto the warehouse plan, where the abscissae belong to the interval $[0, n]$, and ordinates to the interval $[0, m]$. The coordinates of the centres of all individual square units are numbers in the form $k + \frac{1}{2}$ for $k \in \mathbb{N}$.

## Input

The first line of input contains three integers $n$, $m$ and $f$ ($1 \leq n, m \leq 1000$ and $n$ and $m$ are even) indicating the warehouse dimensions and the number of the pillars. Each of the next $f$ lines contains two integers $x_i$ and $y_i$ ($0 \leq x_i \leq n$, $0 \leq y_i \leq m$) denoting the coordinates of the centre of the $i$-th pillar.
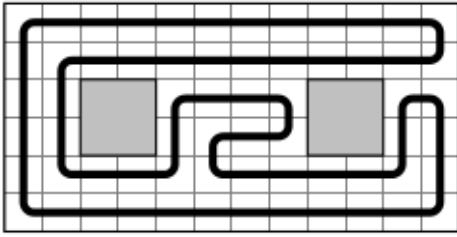
## Output

In the first line of output your program should produce one word "TAK" or "NIE" depending on whether the implementation of floor heating in line with Byteasar's requirements is achievable, or not. In case the answer is "TAK", the second line should contain a description of the exemplary plan of the pipe trajectory in the form of a string of $nm - 4f$ letters. We agree to assume that the beginning of the pipe is located at the point with the coordinates $(\frac{1}{2}, \frac{1}{2})$. Following parts of the pipe are marked as follows: transition by the vector $[0, 1]$ is denoted by a letter 'G', by the vector $[0, -1]$ is denoted by 'D', by the vector $[1, 0]$ is denoted by 'P', and by the vector $[-1, 0]$ is denoted by 'L'. In case there are multiple correct answers, your program should output any of them.

## Examples

| standard input |
|---|
| 12 6 2 |
| 3 3 |
| 9 3 |

| standard output |
|---|
| TAK |
| PPPPPPPPPPPPGGGLDDLLLLLLGPPGLLLDDLLLGGGPPPPPPPPPPPGLLLLLLLLLLLLDDDDD |

## Note

Example output corresponds to the following figure:

# 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 $n$ ($1 \le n \le 500\,000$), denoting the number of years for which the Professor knows the average temperatures. The second line contains a sequence of $n$ integers $t_1, t_2, \ldots, t_n$ ($-10^9 \le t_i \le 10^9$). 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 <br> 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 H. Hit of the Season

| | |
|---|---|
| Input file: | *standard input* |
| Output file: | *standard output* |
| Time limit: | 5.9 seconds |
| Memory limit: | 256 mebibytes |

Byteotian Printing Factory (BPF) has received a large production order for striped wallpaper. Striped wallpaper is the hit of the season in interior design. Each wallpaper has $n$ equal width coloured vertical stripes coloured red, green and blue. BPF is to take care of the design and printing of wallpapers. It is assumed the customer would describe the colour of certain wallpaper stripes. In the case of other stripe colours, the customer allowed BPF a complete freedom.

BPF uses matrices to print a certain number of consecutive bars on the wallpaper. The matrix has a specific colour of each of the printed stripes. The matrix may be shorter than the entire wallpaper. With each application, the matrix stripes must match and overlay the wallpaper stripes; then printing of all the matrix stripes takes place. In this way, a single wallpaper stripe can be printed over more than once. In case a given stripe is printed over using different colours, the final colour is a blend of these colours. The matrix prints only in one orientation and must not in any way be rotated.

BPF employees, irrespective of their sense of aesthetics, would primarily like to design the shortest possible matrix that will allow printing the entire wallpaper. They must bear in mind that in the case of stripes defined by the customer they must use pure colour, without any addition of any other colour. In other words, for each matrix application printing over such a single-colour stripe, the matrix stripe colour must be exactly as defined by the client. No stripe on the wallpaper can remain colourless.

## Input

The first line of input contains one integer $t$ ($1 \leq t \leq 10$) denoting the number of test cases. Each of the next $t$ lines describes a single test case and includes a string of upper-case letters 'R', 'G', 'B' and asterisks ('*'), specifying the desired wallpaper appearance. The letters denote the colour of the stripes, and the asterisks mark the stripes, the colour of which has not been specified by the client. A string is not empty, it consists of a maximum of 3000 characters and contains at most 19 asterisks.

## Output

For each of the test cases your program should output one line containing a string of characters 'R', 'G', 'B': minimum length matrix that allows printing the desired wallpaper. If for a given test case, there are many correct solutions, your program should output any of them.

## Examples

| standard input | standard output |
|---|---|
| 1 | RRGB |
| RRG*R*BRR**B | |

# Problem I. The Staging

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2.9 seconds |
| Memory limit: | 256 mebibytes |

Steven Byteberg is a movie director, specialising in action movies. Presently he is working on a new movie, whose theme is Byteonian Mafia wars. Byteberg wonders what should be the final shape of the movie climax scenes, which will be a spectacular gunfire exchange.

$n$ gangsters participate in this scene, numbered, for simplicity, using consecutive numbers from 1 to $n$. When the tension reaches its critical point, each gangster pulls out his weapon and takes aim at another gangster. Nobody gets aimed at by more than one gangster. Gangsters are poor, but well trained—each can shoot only once, however this shot is always accurate and always deadly.

At some point, one of the thugs cannot withstand the tension any longer and the shooting starts.

The director has determined the initial order in which gangsters have to pull the triggers. Namely, the gangster $i$ shoots towards the gangster $p_i$ at the precise moment $t_i$, unless gangster $i$ already has been killed by that time. The gangster is killed exactly at the same moment when someone shoots in his direction.

The director would like to know how many gangsters will be alive at the end of this scene. Byteberg is not yet completely certain concerning the order in which the gangsters have to shoot. From time to time he commands to change one of the values $t_i$. After every such change he would like to know the number of gangsters who would survive, referring to the new order in which gangsters shoot (taking into account all changes made so far).

## Input

The first line of input contains one integer $n$ ($2 \le n \le 200\,000$), denoting the number of gangsters involved in the scene. The second line contains $n$ integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le n$, $p_i \ne i$, $p_i \ne p_j$ for $i \ne j$) describing whom successive gangsters intend to shoot at.

The third line contains $n$ integers $u_1, u_2, \ldots, u_n$ ($1 \le u_i \le 10^9$), describing the initial order in which the gangsters are shooting: the initial value $t_i$ is equal to $u_i$.

The fourth line contains one integer $q$ ($0 \le q \le 200\,000$), denoting the number of the changes of the values of $t_1, \ldots, t_n$ as planned by Byteberg. Next $q$ lines contain a description of these changes. The $i$-th line contains two integers $k_i$ and $v_i$ ($1 \le k_i \le n, 1 \le v_i \le 10^9$), describing that the $i$-th change consists in setting the value of $t_{k_i}$ to $v_i$. Numbers $u_1, u_2, \ldots, u_n, v_1, v_2, \ldots, v_q$ are pairwise distinct.

## Output

Your program should produce exactly $q + 1$ lines. The first line should contain the number of gangsters who survive the shooting, assuming the initial order of shooting. The $i$-th of the following $q$ lines should present the number of survived gangsters, assuming that the order of shooting is determined by the sequence $t_1, \ldots, t_n$ after performing all the changes—from the first to the $i$-th.

## Examples

| standard input | standard output |
|---|---|
| 4<br>2 3 4 1<br>1 2 3 4<br>3<br>1 8<br>2 7<br>3 6 | 2<br>2<br>1<br>1 |

# 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 $t$ ($1 \le t \le 1000$) 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 $m$ ($2 \le n, m \le 300\,000$), 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 $u_i, w_i$ ($1 \le u_i, w_i \le n$), 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 $a_i$, $b_i$, $d_i$ ($1 \le a_i, b_i \le n$, $1 \le d_i \le 600\,000$). 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 is 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° 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 $n$ ($2 \le n \le 200\,000$) 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 $x_i$, $y_i$ ($0 \le x_i, y_i \le 1\,000\,000\,000$) denoting the coordinates of $i$-th island in the sea. Each island bears different coordinates.
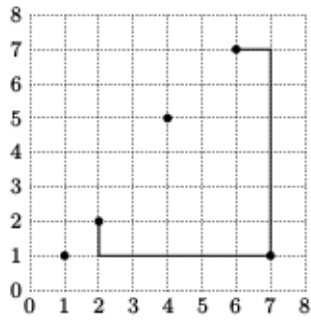
## Output

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<br>2 2<br>1 1<br>4 5<br>7 1<br>6 7 | 2 |

# 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.