Task: BUS
Trams 2

Maratona de Verão, Day 5. Source file `bus.*` Available memory: 128 MB. *24.01.2014*

The recent increases of electricity prices have forced the mayor of Bytetown to take serious budget cuts. In particular, the tram service in Bytetown was suspended. This has caused enormous trouble for the citizens, who now have to use buses. However, there is some hope. Bytetown may receive a special donation from the Ministry of Transport. But for this to happen, they need to analyse the costs of restoring the service.

The tram network in Bytetown consists of n junctions connected with $n-1$ bidirectional track segments. From each junction it is possible to get to any other one. At every junction incident to only *one* track segment, there is a tram terminal. Let us denote the number of terminals by k. The city has $\lfloor k/2 \rfloor$ trams, and each of them has to run along a route that connects two terminals and does not go through any junction multiple times. The routes of distinct trams may go through the same junction or track segment, but at any terminal at most one route may start or end.

The costs of operating a single tram are proportional to the length of its route. Hence, the total money needed to restore tram service depends on the routes of all trams. The mayor would like to know the minimum and maximum costs that may be achieved by considering different routes of the trams. Help him find these two numbers.

Input

The first line contains a single integer n ($1 \leq n \leq 100\,000$) that denotes the number of junctions in the tram network. The junctions are numbered from 1 to n. In the i-th of the following lines contains three integers a_i, b_i, c_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq c_i \leq 1\,000\,000$) that denote the numbers of junctions connected with the i-th track segment and the length of this segment.

Output

Output exactly two lines. The first one should contain the minimum total length of the tram routes that can be obtained by some correct set of routes. The second line should contain the maximum total length of the routes.

Example

For the input data:

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

the correct result is:

```
10
16
```

Task: GEO
Computational Geometry

Maratona de Verão, Day 5. Source file `geo.*` Available memory: 64 MB. *24.01.2014*

Byteasar is preparing the hardest task for the last online round of Algorithmic Engagements 2012. You should not be surprised by the fact that this task is about computational geometry, and more precisely, computing intersections of a polygon and different circles.

Everyone knows that the most difficult part of computational geometry is dealing with special cases. Byteasar has managed to find out that the most evil special case in his task happens when the area of a polygon equals the number of its vertices. Moreover, the edges of the polygon should be parallel to OX or OY axis and every two consecutive edges should be perpendicular. The edges may not intersect apart from pairs of consecutive edges which obviously meet at the same vertex.

Byteasar is having problems with generating such polygons, so he asked you for help.

Input

The first and only line of input contains one integer n ($3 \leq n \leq 10\,000$) denoting the number of vertices of the polygon.

Output

If no polygon with n vertices satisfies Byteasar's requirements, your program should output a single word `NIE` (i.e., *no* in Polish). Otherwise the output should consist of n lines. The i-th line should contain two integers x_i and y_i representing the coordinates of the i-th vertex. The vertices should be given in the same order as they appear on the perimeter of the polygon (the direction along the perimeter in which the vertices are given may be arbitrary). No coordinate may exceed 10^9 by the absolute value.

Example

For the input data:

```
4
```

a correct result is:

```
0 0
2 0
2 2
0 2
```

Task: MIN
Mines

Maratona de Verão, Day 5. Source file `min.*` Available memory: 64 MB. *24.01.2014*

There are some land mines laying on a straight line. When a mine explodes, it detonates all mines within the range of the explosion. For every mine, compute the number of mines that will explode, if the mine is detonated.

Input

The first line contains an integer z denoting the number of test cases that follow.

The first line of each test case contains an integer n ($1 \leq n \leq 100\,000$) specifying the number of mines. Each of the following n lines contains a pair of integers x_i, r_i ($|x_i| \leq 10^{18}$, $0 \leq r_i \leq 2 \cdot 10^{18}$) – the position and the range of explosion of the i-th mine. The mines are given in increasing order of x_i's. No two mines are in the same spot. An exploding mine causes all mines which are *at most* r_i units apart to explode.

Output

For each test case output a single line with n integers $c_1, \ldots, c_n$. The number c_i should be equal to the number of mines that will explode after detonating the i-th mine (including the i-th mine itself).

Example

For the input data:

```
1
5
0 2
2 1
3 2
4 1
6 2
```

the correct result is:

```
4 3 3 3 4
```

Task: NIN
Ninja

Maratona de Verão, Day 5. Source file `nin.*` Available memory: 64 MB. *24.01.2014*

Two Ninja masters, Bytenamura and Bitzuki, have decided to compete in being invisible. The duel will take part in a big forest (which we will view as an Euclidean plane), in which there are n trees (each tree is obviously a point). The ninjas will sneak from one tree to another and try to remain invisible. Hiding behind an infinitely narrow tree is not a problem for them. Each of the masters stars the duel at some tree and his goal is to reach some target tree.

Your task is to find routes for the competitors. Each of the routes has to be a broken line that connects some trees. The two routes cannot intersect or have any common points – if two ninjas meet in some spot, bad things can happen.

Input

The first line contains an integer T that denotes the number of test cases that follow. The first line of each test case contains an integer n ($4 \leq n \leq 60\,000$) – the number of trees in the forest. Each of the n following lines contains a pair of integers x, y ($0 \leq x, y \leq 10^9$) that specify the coordinates of one tree. The first and the second tree are the start and target tree of the first ninja. Similarly, the third and fourth trees are the start and target trees of the second ninja. You can assume that no two trees grow in the same spot and that no three trees lay on a common line.

Output

For every test case output a single line with a word `TAK`, if one can determine the routes for the ninjas. Otherwise, output a line containing a word `NIE`.

Example

For the input data:

```
2
4
0 0
2 2
0 2
2 0
4
0 0
2 0
2 2
0 2
```

the correct result is:

```
NIE
TAK
```

Task: SZP
Spies

Maratona de Verão, Day 5. Source file `szp.*` Available memory: 64 MB. *24.01.2014*

Colonel, here is the city plan: n junctions and m bidirectional roads that connect them. Your task is to transport k spies from junction 1 to junction k. The spies' routes have to be chosen in such a way that not too many spies use the same road. In other words, at most s spies can use each of the roads and the number s has to be as small as possible.

This is a war, not an amusement park. You have to solve the problem for multiple values of k. Dismissed.

Input

The first line of input contains two integers n and m ($2 \le n \le 1000$, $1 \le m \le 300\,000$) that specify the number of junctions and the number of streets in the city. The following m lines describe the streets. Each of them contains a pair of integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) which means that there is a bidirectional street connecting junctions a_i and b_i. There can be at most one road connecting each pair of junctions.

The next line contains an integer q ($1 \le q \le 1\,000\,000$) denoting the number of queries. Each of the following q lines contains a single query. In the i-th of these lines there is an integer k_i ($1 \le k_i \le 1\,000\,000$) which means that k_i spies have to be transported through the city.

Output

Your program should output q lines containing answers to consecutive queries. The answer to a query k_i is the maximum number of spies that will have to use one road in an optimal plan.

If it is impossible to transfer any spies through the city, instead of answering the queries the output should *only* contain exactly 63 lines, each of which contains a word `zelazko`.*

Example

For the input data:

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

the correct result is:

```
1
2
```

*Sorry for that, in Polish it's kind of funny.

Task: TRA
Tram

Maratona de Verão, Day 5. Source file `tra.*` Available memory: 64 MB. *24.01.2014*

Byteman has looked out of the window and seen an old tram at the stop in front of his house. He tried to make a photo of it to add to his collection of old vehicles' photos, but the tram had gone before he reached for his camera.

Byteman lives in Bytetown. There are n junctions in the town numbered from 1 to n and near each junction there is a tram stop. Public transport is very punctual there and the trams arrive at stops exactly at full minutes. Hence, he decided to look out of the window every minute, hoping that the unique tram will appear at the stop again. After a while it became annoying and Byteman decided to look for a smarter solution. He took a map of Bytetown and started analysing it (which was not easy, as he looked out of the window every minute). Eventually, he set the camera to make a photo of the stop every T minutes. He chose the number T as the largest such integer that a camera making photos of the stop every T minutes, starting from the first appearance of the old tram, will not miss any arrival of the tram at the stop, regardless of the route it takes.

The result was so interesting for Byteman, that he started thinking what result (let us denote it by T_j) would he obtain, if he lived next to a different, j^{th}, tram stop. Thus T_j is the largest integer such that if Byteman lived next to the j^{th} junction and the tram appeared at the stop at minute 0, then, regardless of its route, all its following appearances at the stop would occur in minutes which are multiples of T_j. If no track begins at the j^{th} junction or a tram that leaves the junction never returns, $T_j = -1$.

Input

In the first line of the standard input there are two integers n and m ($1 \le n, m \le 100\,000$), separated by a single space, representing the number of junctions in the town and the number of connections between them. The junctions are numbered from 1 to n. In the following m lines there is a description of the tram network in Bytetown. The i^{th} of those lines contains three integers a_i, b_i, c_i ($1 \le a_i, b_i \le n$, $1 \le c_i \le 10\,000$), separated by single spaces. Each triple means that the tram can get from the junction number a_i to junction b_i in c_i minutes. All tracks are one-way, but it can happen that the trams can go both from junction a_i to b_i and from b_i to a_i. It can also happen that $a_i = b_i$ (a balloon loop). There can be at most one connection in a given direction between a pair of junctions.

The time that the tram spends at a stop can be neglected. In addition, we assume that the tram goes as long is it can (until it reaches a dead end or infinitely long, if no dead end is ever reached by it).

Output

Output n integers to the standard output, each in a separate line. The j^{th} line should contain the number T_j.

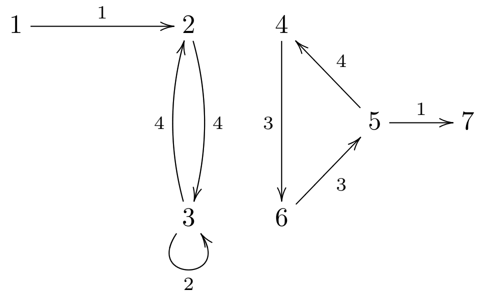
Example

For the input data:

```
7 8
1 2 1
2 3 4
3 2 4
4 6 3
6 5 3
3 3 2
5 4 4
5 7 1
```

the correct result is:

```
-1
2
2
10
10
10
-1
```



Explanation of the example: A tram which leaves junction number 2 can return in, for example, 8, 10 or 12 minutes. Hence, for the camera not to miss any appearance of the tram, it should be set to make photos every two minutes.

Task: ZAG
Puzzle

Maratona de Verão, Day 5. Source file `zag.*` Available memory: 64 MB. *24.01.2014*

Byteasar has bought *Byteotian Times*, a newspaper famous for some original puzzles. One of the puzzles is especially interesting and Byteasar has spent the last few days thinking about the solution.

The puzzle consists in coloring some cells in an $n \times n$ square grid. For each row and column of the grid, we know what is the total number of cells that have to be colored.

Help Byteasar check if it is possible to solve the puzzle.

Input

The first line contains an integer t ($1 \le t \le 100$) denoting the number of test cases that follow.

The first line of each test case contains a single integer n $1 \le n \le 100\,000$) that denotes the dimensions of the grid. The second and third lines contain sequences $w_1, w_2, \ldots, w_n$ and $k_1, k_2, \ldots, k_n$ ($0 \le w_i, k_i \le n$). The number w_i denotes the number of cells that have to be colored in i-th row, whereas k_i – the number of cells that have to be colored in i-th column.

You can assume that the sum of all values of n in all test cases does not exceed $1\,500\,000$.

Output

For each test case, your program should output a single line with a word `TAK`, if the puzzle can be solved, or `NIE` otherwise.

Example

For the input data:

```
2
4
3 2 1 2
1 3 2 2
4
3 2 1 2
1 0 2 2
```

the correct result is:

```
TAK
NIE
```

Task: ZAS
Extension Cords

A group of students went for a training camp in Campinas. Unfortunately, they did not take enough extension cords to plug all their n laptops. After connecting the cords together, they got g sockets to which they can connect their computers.

Each laptop can be characterized by two parameters: the time it can run on a fully charged battery and a time needed to charge an empty battery. The batteries charge with constant speed (independent of their current charge level) and do not use battery power while charging.

The students wonder how long they can work, if they all use their laptops at the same time. In order to maximize this time, they can plug and unplug the laptops from the sockets. We assume that connecting and disconnecting power cords from the sockets takes no time. Initially, the batteries of all laptops are full.

Input

The first line contains two integers n and g ($1 \le n \le 100\,000$, $1 \le g \le 100\,000$). Each of the following n lines contains two integers b_i and c_i ($1 \le b_i, c_i \le 1\,000\,000$), that denote, respectively, the time that one laptop can work on a fully charged battery and the time it takes to charge.

Output

Output a single floating-point number: the maximum time the laptops can work, accurate to at least 5 digits after the decimal dot. You can assume that this time is finite and not greater than 10^{12}.

Grading

Let t be the correct answer and t' be the answer given by your program. Your answer will be considered correct if $\frac{|t-t'|}{t} < 10^{-5}$.

Example

For the input data:

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

the correct result is:

```
3.400000000
```