Problem C. MIPT: Connecting People

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Moscow IPT campus is under renovation. New student accomodations are located in the freshly built skyscraper housing estate. There are n skyscrapers in a row, the i-th from the start of the row having h_i floors. Skyscraper foundations are perfectly level with each other, and all floors have exact same height, thus same floors (counting from the bottom) in any pair of skyscrapers are on the same vertical level.

Each floor of each skyscraper is occupied by exactly one resident (how fancy is that!). Residents can move up and down inside each skyscraper via elevators. Moving one floor in any direction within the skyscraper i takes tv_i seconds.

Really, the only downside of the project is the lack of budget for guarding the entrances to the skyscrapers, thus, for the safety purposes, at the moment entering or leaving the complex is not possible at all. To make up for this, it was decided to build extra corridors connecting skyscrapers in the complex. Each corridor must be perfectly horizontal, thus it must connect floors with same numbers in respective skyscrapers. Further, the corridor can not overlap with any skyscraper standing in between its endpoints. Formally, if floors x are connected with a corridor in skyscrapers i and j, then $h_k < x$ must be satisfied for all k such that $i < k < j$ (and also, naturally, $h_i, h_j \geq x$ must hold).

Corridors are state-of-the-art, thus travelling through any corridor takes th seconds, regardless of the distance travelled. However, they are also expensive, thus only $n - 1$ of them can be built.

To make residents happier (and also complain less about being imprisoned), the following conditions must be satisfied:

- It has to be possible to get from any floor of any skyscraper to any other floor of any other skyscraper via elevators and corridors.
- If we arbtrarily number all residents from 1 to $R = \sum_{i=1}^{n} h_i$, and define $d(x, y)$ as the smallest time (in seconds) the resident x needs to get to the accomodation of the resident y via elevators and corridors, then $\sum_{1 \leq x < y \leq R} d(x, y)$ must the as small as possible.

Help the MIPT planning board to complete this astounding project.

Input

The first line contains two integers n and th — the number of skyscrapers and the time (in seconds) needed to travel any horizontal corridor respectively ($1 \leq n \leq 60$, $1 \leq th \leq 10^6$).

The following n lines describe the skyscrapers. The i-th of these lines contains two integers h_i, tv_i — the number of floors (as well as residents), and the time (in seconds) needed to travel one floor vertically within the skyscraper i respectively ($1 \leq h_i \leq 3000$, $1 \leq tv_i \leq 10^6$).

It is guaranteed that $R = \sum_{i=1}^{n} h_i \leq 3000$.

Output

Print a single integer — the smallest value of $\sum_{1 \leq x < y \leq R} d(x, y)$ over all valid ways to construct $n - 1$ corridors, as defined above.

Examples

standard input	standard output
1 1 5 1	20
2 1 3 3 3 2	59
5 1000 10 1 1 1 7 1 3 1 8 1	460314
5 1 10 1000 1 1000 7 1000 3 1000 8 1000	1626464

Note

In the first sample, there are no corridors, thus the answer is simply the sum of vertical distances.

Optimal configurations for the other sample tests are pictured below:

```
      #           #
      #           #
      #-------#   #       #
      #   #   #   #   #   #
      #   #   #   #   #   #
      #   #---#   #   #   #
      #   #   #   #---#---#
# #   #   # #-#   #   #-# #
#-#   #   # # #   #   # # #
# #   #-# # # #   # #-# # #
```

Problem D. Matryoshka Dolls

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Denisson is an average enjoyer of Matryoshka dolls. A set of matryoshkas consists of a wooden figure, which separates at the middle, top from bottom, to reveal a smaller figure of the same sort inside, which has, in turn, another figure inside of it, and so on.

Recently he bought a set of these toys and placed them on the table in some order. There are exactly n dolls of different sizes in his set, so the current order of matryoshkas can be represented as a permutation p of length n, where p_i is equal to the size of i-th doll.

Denisson usually has a very tight schedule, but today he wants to relax with his toys, and will play the following game:

- Firstly, he will choose some segment $[l, r]$ of his dolls permutation;
- Then he will take the smallest matryoshka on this segment and place it into the matryoshka of the next size. The time needed for him to perform this procedure equals $|i - j|$ where p_i and p_j are sizes of the two smallest dolls on the chosen segment;
- He will repeat this action until there is only one matryoshka left on the segment. The total time needed for his game is the sum of times needed for all performed procedures.

Suddenly, he realized that his interesting game could last for a very long time, but he really cares about his schedule. He came to you with q different segments $[l_i, r_i]$ and wonders what time he needs to play the game on each of these segments. He hopes that you will not spend too much time finding it out.

Input

The first line contains two integers n and q — the number of matryoshkas and the number of requests ($1 \le n \le 10^5$, $1 \le q \le 5 \cdot 10^5$).

The second line describes the order of the matryoshkas represented as a permutation p ($1 \le p_i \le n$).

The following q lines describe requests in the order they are received. Each request is described by two integers l_i and r_i ($1 \le l_i \le r_i \le n$) — endpoints of the i-th segment.

Output

For i-th request, print the total time needed to play the game on segment $[l_i, r_i]$.

Example

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

Problem E. No Rest for the Wicked

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

The 41st Petrozavodsk Programming Camp comes to its end. We hope that you enjoyed the last week. We will probably meet some of you at the ICPC 2020 World Finals, which is planned to be held in about a month, and there are different strategies to spend this time. One option is to train hard and solve more problems. Another possible option is, on the opposite, to clear your mind and take a good rest after all this work.

You may have probably noticed that this camp was held online. COVID-19 has changed the way we live now. It has also constrained travel possibilities a lot.

Let's say that there are n countries (enumerate them from 1 to n for convenience), and there are also m bidirectional flights between them. Each country i has three properties: a spectacularness value s_i, a COVID level c_i and a security threshold $t_i \geq c_i$. Their meaning is the following: if one wants to fly to the j-th country, and they have ever been to the i-th country, then $c_i \leq t_j$ must hold.

Assume that a person from the i-th country wants to visit other countries, and their goal is to go the most spectacular they can — that is, to eventually visit a country j with the maximal possible s_j. Find this spectacularness value for each starting i.

Note that there always is an option to stay home, so the answer always exists. For the sake of simplicity we **do not** require the possibility to return home after visiting the most spectacular possible country (let's say that you can always go home somehow even if there are no flights to it).

Input

The first line contains two integers n and m separated by space ($1 \leq n, m \leq 2 \cdot 10^5$). Then n lines follow, i-th of them contains three space-separated integers c_i, t_i, and s_i ($1 \leq c_i, t_i, s_i \leq 10^9$, $c_i \leq t_i$).

The following m lines describe edges, each of them containing two integers u and v ($1 \leq u, v \leq n$) and denoting a flight between u and v. The described graph is guaranteed to have no self-loops and no multiple edges.

Output

Print n integers, i-th of them being the maximal spectacularness one could see starting from the i-th country.

Example

standard input	standard output
4 3 2 3 1 1 1 4 1 2 2 1 1 3 1 2 1 3 1 4	2 4 2 3

Problem G. Nikanor Loves Games

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Nikanor spends all his free time on games. Because of this, he gets bad marks at the university, but that's another story. He also likes gambling. In this problem, we consider the modification of the game called "Orlyanka". There are two players, and each of them has his own coin. Each of the two sides of a coin contains an integer. Players toss their coins, and the winner is the one with the highest number. We can assume that for each coin the probabilities of coming up both sides are equal.

Tonight Nikanor is playing this game with his friends. Nikanor has n friends, and he will play with each of them for a bet of x_i rubles. Fortunately, Nikanor knows that his i-th friend has a coin with the numbers a_i and b_i. If Nikanor wins against his friend, he will receive x_i rubles. Otherwise, he will pay x_i rubles to his friend. If Nikanor and his friend dropped the same value, Nikanor is declared the winner.

Now Nikanor is going to go to the store and buy one coin for all games to maximize his expected profit, taking the coin cost into account. In this shop, a coin with the numbers a and b costs $a \cdot b$ rubles. Nikanor can buy any coin with positive integers.

It's so hard for Nikanor to make the right decision... Nikanor asks you to help him choose a coin so that the expected profit is as high as possible.

Input

The first line contains one integer n ($1 \le n \le 2 \cdot 10^5$) denoting the number of friends.

Each of the following n lines contains three integers a_i, b_i, and x_i ($1 \le a_i, b_i, x_i \le 10^9$) representing the numbers on i-th friend's coin and i-th bet in rubles.

Output

Print a single integer — the maximum expected profit.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer will be accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

Examples

standard input	standard output
2 1 4 15 3 5 10	2.5000000000
1 2 2 8	4.0000000000

Problem H. Roads of the Empire

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

King is dead. Long live the king!

The young emperor has just inherited the vast empire of his father. After the successful reign, it consists of such many cities that it is plausible to consider that there are infinitely many of them.

However, the transportation system of the empire is pretty poor. As all new rulers do, the emperor wants to change his predecessor's policies. Thus, instead of war, he decides to build some new roads in the empire. However, the country's religious beliefs require the emperor to follow a specific ritual of building new roads.

First, he has to choose a positive number n. Then, n cities are taken, numbered from 1 to n. After that, for all pairs of cities with numbers x and y such that $1 \le x < y \le n$, the road between them is built if and only if $x + n$ is evenly divisible by y.

After all those manipulations, the emperor has to choose two cities u and v, satisfying $1 \le u, v \le n$, and find the number of roads in the shortest path between them. Moreover, he must choose them randomly and equiprobably to be blessed by God, the holy protector of the empire. Knowing this length, the Religious Council will decide whether the plan is acceptable.

The emperor is worried before the meeting with the Council, so he prepared several such plans. For every plan, answer the question of the length of the shortest path!

Input

The first line contains one integer T ($1 \le T \le 2 \cdot 10^5$) — the number of plans proposed by the emperor.

Each of the following T lines contains three integers n, u, v ($1 \le n \le 10^{18}, 1 \le u, v \le n$) — the number of cities chosen for the plan, and the two cities for which you are asked to find the length of the shortest path in the described graph.

It is guaranteed that u and v were chosen randomly and equiprobably after n had been selected for each test.

Output

For every query, print a number on a separate line — the length of the shortest path between the corresponding cities. If the path does not exist, print -1.

Examples

standard input	standard output
4 5 1 2 8 2 5 7 7 2 6 2 5	1 1 -1 2
1 88 14 2	2

Problem I. Drunkards

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

He is a positive man in every way, except that he goes to the bar every evening...

One of your friends is a bartender in the city's most famous (and only) bar. There are $2 \cdot n + 1$ houses in the city, located along one long road and numbered from 0 to $2 \cdot n$. The bar is located in the house numbered n.

The interesting fact is that all drunkards in the city have the same habit. Of course, they leave the bar having a condition not allowing them to go along a way home, so they start to walk without a goal. Namely, every drunkard has an array a in mind, which has length n. In the i-th second after leaving the bar, the drunkard wants to change his position in the road by a_i ($|a_i| = 1$). If the drunkard was in front of the house j, he would be in front of the house $j + a_i$ after this change.

However, they are so drunk that each second with probability $\frac{p}{100}$ they are not capable of moving and stay in their current position.

If a drunkard arrives in front of his house, his family members see him and take him home. Possibly, if he lives in the n-th house itself, his family members will take him immediately. However, after n seconds, if not taken, a drunkard becomes disappointed and sleeps in the street.

Another drunkard came to the bar. The bartender does not know where he lives, so he just assumes for every house the probability is $\frac{1}{2 \cdot n+1}$ that the drunkard lives there. Calculate the probability that his family members will take him home modulo 998 244 353.

Input

The first line contains two integers n and p ($1 \le n \le 5000$, $0 \le p \le 100$), which are described in the statement.

The second line contains n integers $a_1, \ldots, a_n$ ($|a_i| = 1$) — the drunkard's intentions in the i-th second.

Output

Output one integer — the answer modulo 998 244 353.

Formally, let $M = 998\,244\,353$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and it is guaranteed that q is not divisible by M. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q = p \pmod{M}$.

Example

standard input	standard output
2 28 1 1	702764025

Problem J. Rational Dimasik

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Little Dimasik is a rational numbers fan. He has n rational numbers $\frac{x_i}{y_i}$. Recently Dimasik learned how to subtract rational numbers.

Recall that every rational number may be expressed in a unique way as an irreducible fraction $\frac{a}{b}$, where a and b are coprime integers and $b > 0$.

Let us define the function $d\left(\frac{x_i}{y_i}\right)$ as the denominator of the rational number $\frac{x_i}{y_i}$ in irreducible notation. For example, $d(\frac{14}{6}) = d(\frac{7}{3}) = 3$.

Now Dimasik wants to calculate the value

$$\prod_{1 \le i < j \le n} d\left(\left|\frac{x_i}{y_i} - \frac{x_j}{y_j}\right|\right).$$

But soon he realized that this problem is too hard for him. Dimasik asks you to help him. As the value may be very large, find it modulo 998 244 353.

Input

The first line contains one integer n ($1 \le n \le 2 \cdot 10^5$) denoting the number of rational numbers Dimasik has.

Each of the following n lines contains two integers x_i and y_i ($0 \le x_i \le 10^9$, $1 \le y_i \le 10^6$) representing the numerator and denominator of the i-th rational number.

Output

Print a single integer — the answer to the problem modulo 998 244 353.

Examples

standard input	standard output
2 1 3 3 7	21
3 3 2 7 15 5 12	7200

Problem L. Analysing Forests

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Tree is a connected graph without loops (especially, the isolated vertex is the tree too). **Forest** is an union of some non-intersecting trees.

Given the forest of N vertices and M edges. Count the maximal number of the trees in that forest.

Input

First line of the input contains two integers N and M ($2 \le M, N \le 10^4$).

Output

Print one integer — the maximum number of the trees in the forest. If there is no forests for the given M and N, print 0 otherwise.

Examples

standard input	standard output
4 2	2
1201 2020	0

Problem M. Game With A Fairy

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

This is an interactive problem.

You met a fairy in a beautiful forest glade. The fairy is in good mood and would like to give you a present, but not before you best her in a game though.

There are n trunks in the glade; the trunks are numbered from 1 to n. Some of the trunks (**possibly all, and at least one**) contain magical treasure. You can choose several trunks (possibly all) and list their numbers to the fairy. If none of them contain treasure, then you are unlucky and do not get any treasure. However, if more than one of the chosen trunks contains treasure, the fairy thinks you are too greedy and does not give you any treasure either. If **exactly one** of the trunks chosen by you contains treasure, then you get the treasure and are expelled from the forest (that is, the game ends).

You can make at most **200** guesses before the night falls and the fairy becomes annoyed with you. It is guaranteed that the fairy is honest and does not move any treasure after you enter the forest. Can you best the fairy and get the precious treasure?

Interaction Protocol

At the start of the interaction, a single integer n is fed to your program on a separate line. In all test cases except the first one, $n = 10^4$. The first test case coincides with the sample. In each test, the trunks which contain treasure are fixed in advance and stay the same each time a solution is evaluated.

For each query you make, print a single line of n characters; i-th of the characters must be “`1`” if the i-th trunk is chosen for this query, or “`0`” if the i-th trunk is not chosen.

After each query, your program is fed with a line containing “`+`” if you won the treasure, or “`-`” if you didn't win the treasure (note that in this case, you are not told whether no trunks contain treasure, or there are multiple trunks with treasure). After receiving the “`+`” answer, your program must terminate immediately.

Do not forget to end your lines with “new line” characters, and flush your output after each query.

Example

standard input	standard output
3 - - +	100 111 010

Note

In the sample test, magical treasure is in trunks 2 and 3.

Problem N. Portkeys

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

There are n portkeys located on a straight line. Each portkey has three characteristics: x_i (coordinate of the portkey), l_i and r_i. A portkey can be used to teleport to different points on the line. That is, if you are standing at the point x_i, you can use the portkey to instantly teleport to any point y that satisfies $l \le |x_i - y| \le r$. You are not allowed to move along the line without using a portkey. Initially, you are located at the point x_1 (the location of the first portkey).

You are also given m target points on the line. For each target point, find the smallest number of portkeys to be used to get to this point from the start.

Input

The first line contains one integer n, the number of portkeys ($1 \le n \le 2 \cdot 10^5$).

Each of the next n lines contains three space-separated integers x_i, l_i, r_i: the characteristics of the i-th portkey ($-10^9 \le x_i \le 10^9$, $0 \le l_i \le r_i \le 10^9$).

The next line contains one integer m, the number of target points ($1 \le m \le 2 \cdot 10^5$).

The last line contains m space-separated integers y_i: the coordinates of the target points ($-10^9 \le y_i \le 10^9$).

Note that an arbitrary number of portkeys and target points can be located at the same point on the line.

Output

On the first line, print space-separated answers for all target points in the same order the points are given in the input. The answer for a target point is the smallest number of portkeys that are needed to be used in order to get to the point, or -1 if the target point is unreachable via portkeys.

Examples

standard input	standard output
1 1 2 3 4 1 2 3 4	0 -1 1 1
2 0 3 5 5 6 7 5 3 3 12 5 6	1 1 2 1 -1

Problem O. E-bike

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

In the Infinittan city each straight line, parallel to one of the coordinate axis and containing an integer point, is recognized as a street; so each integer point (x, y) is crossing of two streets: horizontal one with the coordinate y and vertical one with the coordinate x.

Dmitry, the system administrator, works in the university, placed near the crossing with coordinates (m, i), and his house is placed near the crossing with coordinates (p, t). Once Dmitry got a chance to test an e-bike. Dmitry knows that for one unit of the way (distance between two neighboring parallel streets) the e-bike use one unit of the power.

Because of high traffic it is impossible to turn (or turn over) anywhere except for the crossing.

Dmitry plans to test e-bike and run from the work to the house starting with fully charged battery, such as at the end of the journey the battery will become empty. Note that Dmitry may visit any crossing (include start and finish crossings) more than once.

Given the battery capacity w check if Dmitry can perform his test.

Input

First line of the input contains two integers m and i — coordinates of the crossing near the university. Second line of the input contains two integers p and t — coordinates of the crossing near the Dmitry's home ($-1000 \le m, i, p, t \le 1000$). Third line of the input contains one integer w — capacity of the battery ($0 \le w \le 10^4$).

Output

In case when Dmitry can do his test, print 'Y'. Otherwise print 'N'.

Example

standard input	standard output
5 4 6 6 5	Y
5 4 6 6 1	N