Pi Duel
Problem ID: piduel

The Math Department is continuing its annual celebration of Pi Day with a pie eating contest organized by the Undergraduate Math Club. Various members of the Math Department will be competing to claim the honor of one of the best pie eaters ever!

"Pie-eating contestants aren't usually asked to contemplate the beauty of a pie's circumference divided by its diameter while devouring as much of it as they can. But on Pi Day, (March 14 - 3.14, get it?), the universal mathematical constant Pi will be celebrated simultaneously by the consumption of its baked edible homonym." - Cornell Math on Pi Day.

You have just entered the final round of the contest, a pie-eating duel, and your opponent is nobody else but the world-famous pie-eating duelist, NumPi, from the Fraternity House of Pi. Two pies were presented to you and your opponent, and the winner would be the person who eats the most pie.

NumPi outmatches you on his pie-eating abilities, but you have the unique advantage of picking which one you would like to eat as the challenger (and the other goes to NumPi). In a glance, you figured out that the two pies have the volume of v_a and v_b respectively; of course, both numbers are prefixes of π (so they are of the form 3.1 or 3.1415 or 3.14159265358979323846264338327950288419716939). With the right choice, you could end up eating more pie than NumPi.

The master and Ph.D. pie-makers of Cornell have made those pies with such precision that both v_a and v_b can be very long. Write a program to decide which one is the right pie to pick!

Input

The input contains two lines, each contains a number which can have at least 2 and at most 1000 digits after the decimal. It is guaranteed that both numbers are prefixes of pi and the two numbers are different.

Output

Print $first$ or $second$, the pie with more digits of π.

Sample Input 1	Sample Output 1
3.14 3.14159	second

Slope Day Countdown
Problem ID: slopedaycountdown

Slope Day is an annual day of celebration held at Cornell University the Thursday after the last day of undergraduate classes, and it includes live music and catered food.

Slope Day is coming, and you just can't wait to see your favorite artists play. You don't want to miss a thing! To help you be the very first person to arrive, you set up a countdown timer that shows how many seconds are there until Slope Day starts. However, as you quickly realized, displays like "314159 seconds" aren't very intuitive. So you decide to upgrade it to a more human-readable version. The logic you have in mind can be described as follows:

If there are 0 seconds left, the program should print "NOW".

If there are fewer than 60 seconds left, the program should print "s␣seconds" where s is the number of seconds left without leading zeros.

If there are fewer than 3600 seconds left, the program should print "m␣minutes␣s␣seconds" where m is the number of minutes left and s is the number of seconds left after m minutes.

If there are fewer than 86400 seconds left, the program should print "h␣hours␣m␣minutes␣s␣seconds" where h, m, and s are the number of hours, minutes, and seconds left respectively.

Otherwise, the program should print "d␣days␣h␣hours␣m␣minutes␣s␣seconds" where d represents the number of days left and other variables the same as above.

Input

The input contains single line with a single integer T, $0 \le T \le 10^9$, the number of seconds before Slope Day.

Output

Print a single line in the format described in the problem statement. DO NOT print extra whitespaces.

Sample Input 1	Sample Output 1
`59`	`59 seconds`

Sample Input 2	Sample Output 2
`3603`	`1 hours 0 minutes 3 seconds`

Sample Input 3	Sample Output 3
`0`	`NOW`

Dragon Day Parade
Problem ID: dragonparade


This was 2019's dragon. A new dragon is designed and built every year.

"Every year in March, in a tradition that goes back more than 100 years, an enormous dragon created by first-year architecture students parades across campus. Accompanied by AAP students in outrageous costumes, the dragon lumbers to the Arts Quad where it does battle with a phoenix created by rival engineering students. This rite of spring is one of Cornell's best-known traditions." - Cornell AAP on Dragon Day.

This year, the students have built an enormous dragon of N segments on Campus Rd, which can be considered an infinitely long straight line. The segments are numbered from 1 to N.

As one of the organizers, you are given the challenging task of assigning the team of exactly N students to the N segments of the dragon.

Each student must be assigned to a unique segment. Student i is at the position P_i at time 0, and the student assigned to segment j must reach position Q_j to operate segment j.

All students can travel at the speed of a unit of distance per second in either direction, and they move in parallel without interfering with one another.

What is the minimum time for all students to reach their posts and start marching to the Arts Quad?

For example, suppose the dragon has $N = 2$ segments that should be on positions $Q_1 = 1865$ and $Q_2 = 2012$, and the students are initially at positions $P_1 = 2000$ and $P_2 = 1900$.

There are two ways to assign students to dragon segments:

- Student 1 is assigned to segment 1 (and student 2 to segment 2). Student 1 will take $2000-1865 = 135$ seconds, and student 2 will take $2012 - 1900 = 112$ seconds, so the total time would be 135.
- Student 2 is assigned to segment 1 (and student 1 to segment 2). Student 1 will take $2012 - 2000 = 12$ seconds, and student 2 will take $1900 - 1865 = 35$ seconds, so the total time would be 35.

Therefore, the correct solution is 35, as the second option is better.

Input

The input starts with a single line containing a single integer N, $(1 \leq N \leq 10000)$, the number of segments and the number of students.

The second line contains N integers, P_i, $(1 \leq P_i \leq 10^5)$, the position of the students at time 0.

The third line contains N integers, Q_i, $(1 \leq Q_i \leq 10^5)$, the required position the segments.

It is NOT guaranteed that the position of the students or the required position of the segments would be given in any specific order.

Output

Print a single integer, T, the minimal time for all students to reach their assigned post for some optimal assignemnt.

Sample Input 1	Sample Output 1
2 2000 1900 1865 2012	35

Sample Input 2	Sample Output 2
1 1000 1000	0

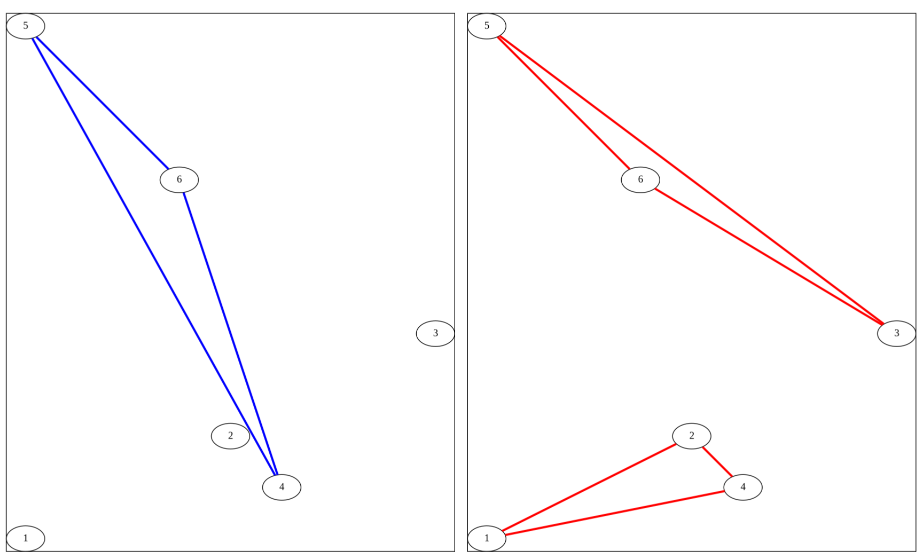
Far Above Cayuga's Water
Problem ID: abovecayuga

> "Far above Cayuga's waters,
> With its waves of blue,
> Stands our noble Alma Mater,
> Glorious to view."
>
> - Far Above Cayuga's Water

"Far Above Cayuga's Waters" is Cornell University's alma mater. The lyrics were written circa 1870 by roommates Archibald Croswell Weeks (Class of 1872) and Wilmot Moses Smith (Class of 1874) and were set to the tune of "Annie Lisle," a popular 1857 ballad by H. S. Thompson about a heroine dying of tuberculosis.

Knowing this song well, Alice, the brave Queen-Knight, wonders what is really there far above Lake Cayuga's water. So she looked at a satellite map and found many points of interest, such as the Herbert F. Johnson Museum of Art, the Cornell Botanic Gardens, the Bloomberg Center of Cornell Tech, and even the new Cornell Ann S. Bowers College of Computing and Information Science building (which is mostly occupied by the department of post-quantum machine learning). What is even more curious about this image is that she could draw many triangles out of those points! Maybe triangles are what's really there?

Given N points on a 2D plane, what is the maximum number of triangles with positive area one can draw using those points as vertices and using each point at most once?



The blue subfigure forms one triangle (note that you can't form another triangle with the three remaining points) while the red one forms two.Two is the maximum possible number of triangles, so the answer is 2. This drawing reflects sample input 1 below.

Input

The first line of the input contains a single integer N, $1 \leq N \leq 10^5$, the total number of points.

Each of the following N lines contains two integers x, y, $1 \leq x, y \leq 10^9$, the coordinate of a point of interest. The input guarantees that all coordinates are different.

Output

Output a single integer, the maximum number of triangles one can draw under the restriction.

Sample Input 1	Sample Output 1
6 1 1 5 3 9 5 6 2 1 11 4 8	2

Sample Input 2	Sample Output 2
8 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8	0

RBG
Problem ID: rbg

One of Cornell's new North Campus dorms will be named in honor of the associate justice of the Supreme Court of the United States, Ruth Bader Ginsburg '54. This problem is also named after her.

You are designing a virtual reality (VR) game simulating a supreme court hearing. In particular, you are now creating the method to render the courtroom, and for that you need to decide the color of each pixel of the VR display.

The courtroom can be described as a box which is W points wide, L points long, and H points high. Each point is described by its color in red-blue-green space "RBG" (the *legislative* variant of the more well-known RGB). In other words, the color of each of the $W \times L \times H$ points will be divided in 3 components (red, blue, and green). Each component will be a number between 0 and 255.

For instance, the color red is defined as $(255, 0, 0)$, while yellow is $(255, 0, 255)$ (combination of equal parts red and green).

A particular pixel to be rendered represents a box within the room. For instance, a pixel might represent the box $(1, 3, 5) - (3, 10, 7)$, which means that all points (x, y, z) with $1 \leq x \leq 3$, $3 \leq y \leq 10$, and $5 \leq z \leq 7$ are represented by that pixel. A pixel should be colored with the *average* color of the points represented by it!

For instance, if a pixel represents points with color $(10, 0, 255)$, $(50, 50, 50)$, and $(255, 0, 0)$, then the pixel should have color $(\frac{10+50+255}{3}, \frac{0+50+0}{3}, \frac{255+50+0}{3}) = (105, 16, 101)$ (note that all divisions should be rounded down).

You are given the color of all the points of the courtroom and a sequence of pixels, and it's your job to decide the color of each pixel.

Input

The first line of the input contains three integers W, L, H, with $1 \leq W$, $1 \leq L$, $1 \leq H$ and $W \times L \times H \leq 10^5$.

Next you will receive W blocks of lines, each block will have exactly L lines, and each line will have exactly $3 \times H$ integers between 0 and 255.

The interpretation is that the first three numbers correspond to the colors (in RBG) of point $(1, 1, 1)$, then point $(1, 1, 2)$ will be given, and so far so on (until $(1, 1, H)$, followed by $(1, 2, 1)$ then $(1, 2, 2)$, until $(1, 2, H)$, $\ldots$). Please see the samples below to understand the input format.

The next line will contain a single integer Q ($1 \leq Q \leq 10^5$) with the number of pixels to be drawn.

The next Q lines will contain 6 integers in $x_1, y_1, z_1, x_2, y_2, z_2$ (with $1 \leq x_1 \leq x_2 \leq W$, $1 \leq y_1 \leq y_2 \leq L$, $1 \leq z_1 \leq z_2 \leq H$) with the description of the box of points that this pixel represents (it represents a total of $(x_2 - x_1 + 1) \times (y_2 - y_1 + 1) \times (z_2 - z_1 + 1)$ points).

Output

The output should contain Q lines, one for each pixel.

The i-th line should contain 3 integers with the color of the i-th pixel in RBG.

(Read the sample inputs below for further clarification)

Sample Input 1

```
1 2 3
0 1 2
3 4 5
6 7 8
9 10 11
12 13 14
15 16 17
3
1 1 1 1 2 3
1 1 1 1 1 1
1 2 2 1 2 3
```

Sample Output 1

```
7 8 9
0 1 2
13 14 15
```

Pumpkin on the Clocktower
Problem ID: pumpkinclocktower

One of the greatest mysteries in Cornell history is about a pumpkin placed atop the 173-foot-tall spire of McGraw Tower in October 1997. It has since been memorialized as an ice cream flavor, “Clocktower Pumpkin,” produced by the Cornell Dairy.

A sixty pounds pumpkin just appeared on top of McGraw Tower

No one has ever figured out how the pumpkin got up there, but one hypothesis that involves temporary support beams has caught our attention.

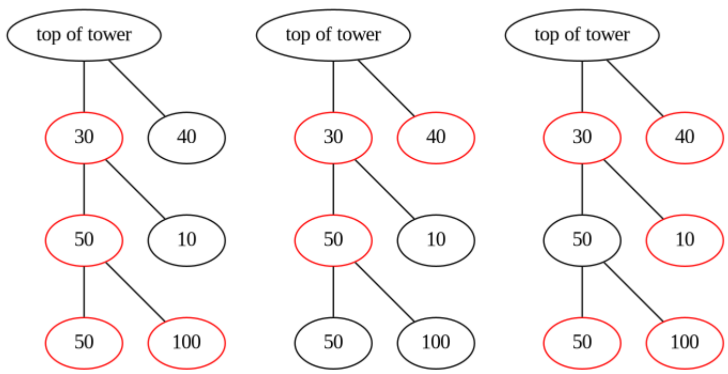
A close examination of the clock tower suggested that there are N positions where beams can be placed.

Each possible position is close to exactly one position above it (and may be close to zero or more positions below it). Such a close position can be either the top of the tower or another possible position for a beam.

In other words, if we connect each possible position to the position above it that is close to it, we will get a tree with $N+1$ nodes and N edges, with the top of the tower as its root.

Each support beam i would provide some steadiness value s_i if placed. The civil engineering department advised you that if a beam is placed, then at most 1 other beam that is close to it can be placed.

To test the hypothesis of placing the pumpkin through temporary support beams, you wonder what would be the maximum possible sum of steadiness values of a proper support beam placement.



Sample input 1:

The first selection of beam positions has value 230 but is invalid because a beam has 3 other beams close to it.
The second one is valid, and has value 120.
The third one is also valid, and has value 230, which is the highest possible.

Input

The first line of input contains a single integer N, $1 \le N \le 10^5$, the total number of possible positions. Positions are numbered from 1 to N, and the top of the tower is numbered 0. Positions are listed in decreasing order of height. So position 0 is the highest (top of the tower) followed by position $1, \ldots$

The following N lines describe the tree structure. The i-th line contains two integer p_i and s_i ($0 \le p_i < i$ and $1 \le s_i \le 10^4$), which are the position above i which is close to it and the steadiness of a beam placed at position i respectively.

Output

Output a single value, the maximum possible sum of steadiness values.

Sample Input 1

```
6
0 30
0 40
1 50
1 10
3 50
3 100
```

Sample Output 1

```
230
```

Sample Input 2

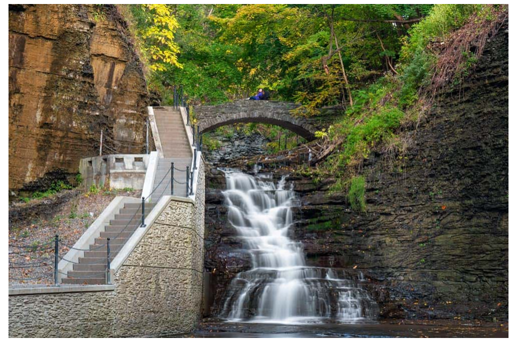
```
10
0 10
0 10
0 10
0 10
0 10
0 10
0 10
0 10
0 10
0 10
```

Sample Output 2

```
100
```

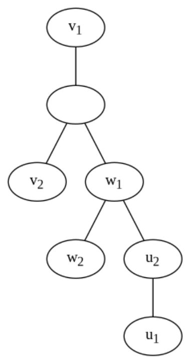
Picnic in the Gorges
Problem ID: gorgepicnic

Over the summer and at the beginning and end of the academic year, when Ithaca enjoys warmer weather, Cornell students are known to find respite from the heat in the two creeks that cut across Cornell's Ithaca campus through two dramatic gorges — Cascadilla and Fall Creek.



Surprisingly, whenever a waterway divides in two, these two branches never meet again. In other words, the waterfall and creeks form a tree with waterfalls as nodes and waterways as edges. The waterfalls are conveniently numbered from 1 to N. Two Cornell students, Hiro and Sophie, are planning a picnic in the beautiful gorges over the weekend. Hiro will start from waterfall u and Sophie from a different waterfall v. They will each then take the shortest route along the water to meet at a third waterfall w for lunch.

They wonder whether they can meet somewhere before waterfall w and walk the rest of the way enjoying each others company.



In case 1 above (Hiro is at u_1, Sophie at v_1, and they want to meet at w_1), they will only meet at w_1.
But in case 2 they can meet before reaching w_2 (namely at the node labeled w_1).

This example corresponds to sample input 1 below.

Input

The first line of the input contains a single integer N, $3 \leq N \leq 10^5$, the number of waterfalls around Cornell.

The following $N-1$ lines describe the waterways. Each line contains two integers a, b, $1 \leq a, b \leq N$ indicating waterfall a and waterfall b are directly connected by a creek. The input guarantees the waterfalls and creeks form a valid tree.

The $N+1$-th line contains a single integer Q, $1 \leq Q \leq 10^5$, the number of days when Hiro and Sophie would like to have lunch.

The following Q lines describes each day. Each line contains three integers u, v, w, all between 1 and N, and all different describing where Hiro starts that day, where Sophie starts, and where they want to meet.

Output

For each query, output a separate line of "YES" (without quotes) if Sophie and Hiro might run into each other on their way from u to w and v to w; output "NO" otherwise.

Sample Input 1	Sample Output 1
7 1 3 2 3 3 4 4 5 4 6 6 7 2 7 1 4 6 2 5	NO YES

The Final Touchdown
Problem ID: finaltouchdown

Attention: This problem is HARD! Only attempt to solve it after you are done with the other problems.

Touchdown, or the Big Red Bear, is the unofficial mascot of Cornell University. It appears on the logo for Cornell Athletics and is represented in a statue erected outside Teagle Hall since 2015.

It's the year 2021. Your team is participating in the Cornell University High School Programming Contest, and you have come this far. The only problem left between your team and the ultimate victory of solving all problems in the contest is this problem called "The Final Touchdown." All you need to do is come up with some magical string, usually regarded as "the source code," typing it into a text box and hitting the submit button. Almost too easy - except that you have no clue of what that magical string look like.

Fortunately Touchdown comes to help you win the game! He gives you the following clues:

- The solution you are looking for is a permutation of the numbers from 1 to N.
- Touchdown tells you that M of the N numbers are red (the others are white). He also tells you which are these numbers.
- Touchdown tells you a strictly increasing sequence S of K numbers between 1 and N, and he guarantees that S is a subsequence of the solution. Furthermore, no increasing subsequences in the solution has more red numbers than S. (Touchdown calls S a *Big Red Sequence*)

But there could still be many such permutations. Write a program to compute the number of permutations that satisfy Touchdown's description. As the number might be very large, output it modulo $10^9 + 7$.

Input

The first line contains three integers N, M, K, $1 \leq N \leq 200$, $0 \leq M \leq min(15, N)$, $1 \leq K \leq min(50, N)$.

The second line contains M integers, the numbers that are colored red.

The third line contains K integers in strictly increasing order, the Big Red Sequence.

Output

Output a single integer, the number of permutations consistent with what Touchdown told you.

Sample Input 1	Sample Output 1
5 4 3 1 2 3 4 1 3 4	15

Sample Input 2	Sample Output 2
10 5 1 1 3 5 7 9 5	30240