11th Virginia Tech High School Programming Contest

Dec 14, 2024

As a reminder, here are the key rules under which this contest is conducted:

- Teams may not communicate with another human during the contest about the problems.
- Teams may not use more than 1 computer.
- AI assistance (CoPilot, TabNine, ChatGPT, etc.) is not permitted.

This problem set contains 12 problems which target a variety of skill levels. You are not expected to solve all of them, particularly if this is your first programming contest! The problems are not sorted by difficulty - make sure you peek at all of them and have a look at the scoreboard to spot problems that may be easier than others.

Enjoy!

This page is intentionally left blank.

Problem A
Check Your Color Set

A image of a color palette containing 12 different colors
Source: By G4sxe at the English Wikipedia

As a hobbyist graphic designer, spence often finds it fascinating to observe how colors interact while designing. Sometimes, the colors don't complement each other, and determining when this happens can feel like guesswork. Fortunately, the graphics design community has a metric that allows spence to decide how far two colors are from each other, and this metric is called the color difference.

When designing posters, spence represents each color as a point in 3D space, where the red, green, and blue (RGB) components correspond to the x, y, and z coordinates. To evaluate the contrast between two colors, spence computes the Euclidean distance between their respective points. A color difference of at least 128 ensures that the colors provide enough visual differentiation.

Your task is to help spence develop a program to verify whether the colors in his poster designs meet the minimum contrast threshold for all pairs of colors.

Input

The input starts with a single integer n ($1 \le n \le 100$), the number of poster designs (color sets). This is followed by n descriptions of poster designs. For each poster design, the first line contains an integer m ($2 \le m \le 20$), the number of colors used in the poster. The following m lines each contain three integers R, G, and B ($0 \le R, G, B \le 255$), representing the RGB values of a single color.

Output

The output should consist of n boolean values, each on a new line. Each boolean indicates whether all color pairs in the corresponding poster meet the minimum contrast requirement. Output `True` if all pairs in the poster meet the threshold; otherwise, output `False`.

Sample Input 1

```
2
3
0 0 0
255 255 255
128 128 128
2
0 0 0
50 50 50
```

Sample Output 1

```
True
False
```

This page is intentionally left blank.

Problem B
Communication Bases

Image by Pixabay from Pixabay

In a futuristic world, humans are at war with aliens. There are N human bases numbered 1 to N, and they are spread throughout the galaxy. Humans have set up M bi-directional communication links between these bases, but some links have sustained damage from alien attacks. This has separated the human bases. You are given a list of communication links, and a score, s, for each of these links.

This score s represents the amount of resources required to maintain the link. The higher the value of s, the more resources are needed to keep the link operational. While lower-resource links are cheaper to maintain, they tend to be less reliable under the current war conditions and are prone to breakdowns.

The human military has decided to prioritize reliable and resource efficient communication links. For this strategy to succeed, exactly K connected components of bases are required. Therefore, any link with a resource score $s < T$ is considered too unreliable and must be deactivated, even if it is cheaper. Your task is to find the smallest threshold T such that there are exactly K connected components.

A connected component is a subset of bases such that for each pair of bases in the subset, there is a path involving one or more reliable links that leads from one base to the other. A single base that is not connected to any other base also forms a connected component.

Input

The input begins with a single line containing three space-separated integers: N, M, and K, where: N represents the number of human bases ($1 \leq N \leq 10^5$), M denotes the number of communication links ($1 \leq M \leq 10^6$), and K is the number of components required ($1 \leq K \leq N$).

The next M lines each contain three integers, u, v, and s, where: u and v indicate a bidirectional communication link between base u and base v, and s is the score of the communication link ($1 \leq u, v \leq N$ and $0 \leq s \leq 10^5$). There will be at most one link between any two pairs of bases.

Output

The output consists of one integer T, the minimum threshold such that there are exactly K connected components of bases. If no threshold, T, allows the graph to be partitioned into exactly K connected components, output `IMPOSSIBLE`.

Sample Input 1	Sample Output 1

5 4 3
1 2 6
2 3 4
3 4 8
4 5 10
	7

Sample Input 2

```
8 7 4
1 2 6
2 3 5
3 4 8
4 5 10
6 8 30
7 1 2
7 8 2
```

Sample Output 2

```
6
```

Problem C
Hokie Parade

At Virginia Tech's annual homecoming parade, a line of Hokies is marching down Main Street. Each Hokie has a "spirit score" represented by an integer (which can be positive, zero, or negative).

Image by Gary Cope.

The parade organizers want to create a spectacular formation by splitting the line into two consecutive non-empty groups where the total spirit score of each group is equal.

Your task is to determine the number of ways the line can be split between two Hokies so that the sum of the spirit scores in the first group is equal to the sum in the second group.

Input

The first line contains an integer N ($2 \leq N \leq 5 \cdot 10^5$), which is how many Hokies are in the line. The next line contains N integers s_i ($-10^9 \leq s_i \leq 10^9$), which represent the spirit score for each hokie.

Output

Output a single integer that represents the number of ways the line can be split into two consecutive non-empty groups with the sums of their spirit scores equal to each other.

Explanation

For Sample Input 1, the parade can be split in three ways. In between position 3 and 4, 6 and 7, and 7 and 8.

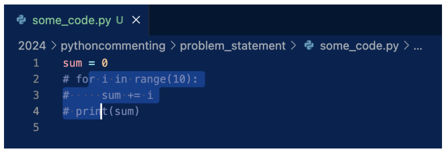
Sample Input 1	Sample Output 1
9 1 5 -6 7 9 -16 0 -2 2	3

Sample Input 2	Sample Output 2
3 1 1 1	0

This page is intentionally left blank.

Problem D
Python Commenting

There exists an IDE and text editor named Visual Studio Code (which you might be using right now). When using it to edit a file, if you select the lines of text in the closed interval $[L, R]$ and press `Ctrl+/`, one of the following two things happens:


some_code.py U
2024 > pythoncommenting > problem_statement > some_code.py > ...

```
sum = 0
# for i in range(10):
#     sum += i
# print(sum)
```

Commenting out some Python code in VS Code.

1. If at least one of the lines selected has not been commented out already, then each of the lines from L through R add one comment indent, represented by a new “`# `” prefixing each of those lines.
2. If all of the lines selected have been commented out at least once, then each of the lines from L through R remove one comment indent, represented by the “`# `” at the beginning of those lines being removed.

For example, with the following code:

```
1 sum = 0
2 for i in range(10):
3     sum += i
4 print(sum)
```

If you were to select lines 2 through 4 and comment them out, then you would have:

```
1 sum = 0
2 # for i in range(10):
3 #     sum += i
4 # print(sum)
```

After another operation with $[L, R] = [1, 3]$:

```
1 # sum = 0
2 # # for i in range(10):
3 # #     sum += i
4 # print(sum)
```

And then with one more operation with $[L, R] = [3, 4]$:

```
1 # sum = 0
2 # # for i in range(10):
3 #     sum += i
4 print(sum)
```

You are given some code with a valid sequence of operations of type 1 having been applied but none of type 2 having been applied. Calculate the minimum number of commenting operations needed to produce the given code, assuming the code originally started out with no comments.

Input

The first line of input gives n ($1 \leq n \leq 10^4$), the number of lines of code present. The next n lines all have `some code` on them (literally), with some (possibly no) instances of “`# `” at the beginning of them.

Output

Print out the minimum number of commenting operations needed to produce the given code.

Sample Input 1

```
3
some code
some code
some code
```

Sample Output 1

```
0
```

Sample Input 2

```
4
# some code
# # some code
# # some code
# some code
```

Sample Output 2

```
2
```

Sample Input 3

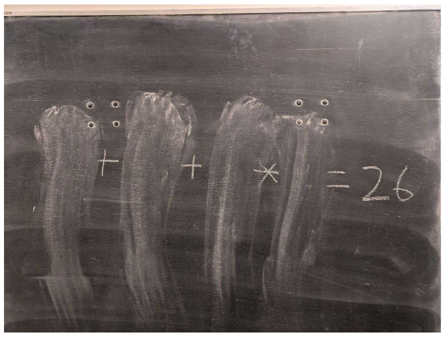
```
13
some code
# some code
# some code
# # # some code
# # # some code
# # some code
# # some code
# # some code
# some code
some code
# some code
# # some code
# some code
```

Sample Output 3

```
5
```

This page is intentionally left blank.

Problem E
Erased Numbers



equation on a blackboard with numbers erased

Alice walks into binary math class late one day but it was too late. Somebody had erased all the numbers on the blackboard! All that is left now are the addition and multiplication operators and the final number the mystery equation is equal to. The class is binary math class so all numbers are some power of two. She knows that there can be multiple possible equations so with no better option, she will guess one of the possible ways to fill in the equation. Help her calculate how many possible answers there are so that she can at least impress her teacher so that he might possibly not deduct all of her points. Since the number might get very large, compute the remainder of that number when divided by $10^9 + 7$.

The equation left on the blackboard may look like the one below where `?` could be any nonnegative power of two.

```
? * ? + ? = 3
```

In the example above, there are 3 possible ways to fill in the `?`, namely,

```
2 * 1 + 1 = 3
1 * 2 + 1 = 3
1 * 1 + 2 = 3
```

Note:

- All `?` are integers of the form 2^p where p is an integer with $p \geq 0$.
- $10^9 + 7$ is a prime number.

Input

The first line contains two integers, n and k ($1 \leq n \leq 100, 1 \leq k \leq 10^5$), where n is the number of `?` in the equation given in the next line and k is the number the left-hand side of the equation is equal to. The second line is a string s of length $2n - 1$ composed of n question marks (`?`) and $n - 1$ arithmetic operators, each of which is `*` or `+`.

Output

Output a single nonnegative integer $p \pmod{10^9 + 7}$ where p is the number of different ways to fill in the question marks in the equation and get number k.

Sample Input 1

```
3 3
?*?+?
```

Sample Output 1

```
3
```

Sample Input 2

```
5 10
?+?*?+?*?
```

Sample Output 2

```
30
```

Sample Input 3

```
5 5
?+?*?+?*?
```

Sample Output 3

```
8
```

Sample Input 4

```
1 1
?
```

Sample Output 4

```
1
```

Problem F
Wearing Jackets

Jacket Motorcycle Protection Suit, Source: PixaBay

You are not a big fan of temperature changes. You always prefer the temperature to be 70 degrees Fahrenheit. Unfortunately, the temperature where you live is typically lower. To combat this, you purchased K jackets that each individually increase your temperature by T_i. They don't fit on you when you try wearing more than one at a time, and you don't have enough room to carry any additional jackets throughout the day, so you can only bring up to 1. At any time during the day, you can take off your jacket or wear the jacket you brought, if you brought one. You just checked the forecast for the next D days and found the high and low temperatures, L_j and H_j, for those days. For every integer temperature from L_j to H_j, there will be a time when this is the current temperature. You want to keep the temperature you feel as close to 70 degrees at all times.

Minimize the largest temperature difference from 70 degrees you will feel each day by optimally choosing what jackets you'll wear (if any).

Input

The first line of input contains K and D, each integers between 1 and 1 000. The next K lines each contain an integer with a temperature increase for one jacket. Line number i will have the temperature increase d_i for the i^{th} jacket, which is an integer between 1 and 70 degrees Fahrenheit. The next D lines after that will have the low and high temperatures for the day. These temperatures will be given in Fahrenheit and be integers between 0 and 70.

Output

On D lines, output a nonnegative integer representing the optimal temperature difference for each day between day 1 and day D.

Sample Input 1	Sample Output 1

3 2
1
3
5
63 68
66 70
	2
	1

This page is intentionally left blank.

Problem G
Multithreading

Source: https://chatgpt.com/

Dr. Back has recently noticed that Sri has shown up late to his Computer Systems class multiple times. Sri keeps saying he'll stop doing it on *this day* or *that day*, but it hasn't happened yet. In order to show him just how bad of an idea this is, Dr. Back decided to run an experiment. Instead of teaching a lesson on multithreading, he has decided to perform multithreading on reality itself: he will split every one of these hypothetical end dates into a new timeline and examine them individually.

Let each class throughout the semester have some associated amount of knowledge that Sri would obtain during that class if he didn't show up late for that class or any future classes. The issue with showing up late for a given class is that the knowledge gained from the previous classes (or the later classes, depending on how you interpret it) is effectively decreased, since the material builds on top of itself. So, if, in some alternate timeline, Sri says that he will keep showing up late until some certain day and then never do it again after that, then all of the knowledge values until and including that day would surely become limited in that timeline as an effect of Sri being late to, and taking in less material from, his classes, which occasionally (luckily for him) causes no decrease at all.

Specifically, given that there are n classes in the semester, we will have a knowledge array k of length n containing the knowledge values from each class. If Sri says that he will keep showing up late until class c_{end}, then for the timeline where that is true, we will have a new knowledge array k_{new} with the following modifications: all of the days after c_{end} will be unaffected (i.e., $k_{new_c} := k_c$), and for each class $c \in \{1, 2, \ldots, c_{end}\}$, k_{new_c} will be set to the minimum knowledge across all knowledge values for the classes between the current class c and c_{end}, inclusive (i.e., $k_{new_c} := min(k_c, k_{c+1}, k_{c+2}, \ldots, k_{c_{end}})$).

By creating a new timeline for each c_{end} value that Sri whimsically threw out, Dr. Back can analyze the k_{new} array corresponding to that timeline and show Sri how much it reduces the amount he learns; he does this by simply computing the total knowledge, t, gained in that timeline (i.e., the sum of all knowledge values in k_{new}). Your task is to find t for each given c_{end}.

Input

The first line contains an integer n ($1 \leq n \leq 200,000$), which is the length of the knowledge array. The second line contains n integers, which are the elements of the knowledge array k ($1 \leq k_c \leq 10^9$) with spaces in between them. The third line contains an integer q ($1 \leq q \leq 200,000$), the number of different c_{end} values Sri has stated to Dr. Back. The fourth line contains q space-separated c_{end} values ($1 \leq c_{end} \leq n$).

Output

For each c_{end}, output the corresponding t. The outputs for each query should be separated by spaces.

Sample Case 1 Explanation

If Sri shows up late for the last time on day 4, then k_{new} would look like $[2, 4, 4, 4, 7, 5]$, whose sum is 26. Similarly,

$c_{end} = 2 \Rightarrow k_{new} = [2, 4, 5, 4, 7, 5] \Rightarrow t = 27$

$c_{end} = 6 \Rightarrow k_{new} = [2, 4, 4, 4, 5, 5] \Rightarrow t = 24$

$c_{end} = 3 \Rightarrow k_{new} = [2, 4, 5, 4, 7, 5] \Rightarrow t = 27$

Sample Input 1

```
6
2 4 5 4 7 5
4
4 2 6 3
```

Sample Output 1

```
26 27 24 27
```

Sample Input 2

```
12
5 3 4 3 2 9 7 5 3 9 5 6
6
6 2 12 3 1 4
```

Sample Output 2

```
54 59 38 59 61 58
```

Problem H
Remaking Orders

A delicious bowl of food
Source: Pexels

You work in a very busy dining hall making food bowls that can have many different ingredients. When an order comes in, you read the list of ingredients to add to the bowl, add them one by one, and move on to the next order. However, every once in a while, you add the wrong ingredient to a bowl by mistake. Since you want to make sure that everyone gets their order correctly, you put the bowl aside and make a new one.

Sometimes, you realize that one of the next customers just so happens to want all of the ingredients that you had added to the bowl you set aside. In those cases, you can take that bowl again, add any other ingredients that may be missing, and serve it. For this reason, when you screw up an order, you always hold on to that bowl and try to use it later on.

After a while, the bowl will get cold, so if you haven't managed to reuse it, you have to throw it away. You know that the bowl will be too cold to use if you have made a certain amount of bowls after setting it aside and you still haven't used it. However, when you reuse a bowl that you had set aside and add new ingredients to it, the new ingredients will heat it up again and should you later make a mistake, you can keep the bowl for the same number of orders as when you added ingredients to it the first time.

On your counter, you only have space for two bowls: the one you're making right now and at most one that you set aside. For this reason, when you screw up a bowl and you already have a bowl set aside, you have to throw the old one away and keep the new one on the counter. At the end of the day you must throw out any unfinished bowl.

You are curious about how many bowls you throw away each day, and you want to write a program to find out. However, all you have is the list of orders and a log of each time you added an ingredient to a bowl.

Input

Input begins with a line containing 3 integers, m, n and c. m ($1 \leq m \leq 1\,000$) is the number of ingredients available to order, which are numbered from 1 to m. n ($1 \leq n \leq 1\,000$) is the number of orders received that day. c ($0 \leq c \leq 1\,000\,000$) indicates how many complete bowls you can make after you last set aside away a bowl before it gets cold. For example, if $c = 2$, you can make 2 bowls without using the one you set aside, but if you don't use it on the third bowl, it will get cold and will have to be thrown away. If you start using it on an order and then screw up, you can set it aside and make another 2 bowls.

The following n lines contain all the orders for that day. Each line begins with an integer i ($1 \leq i \leq m$), the number of ingredients in the order. This is followed by a list of ingredients in the order, separated by spaces. All ingredients will be numbers in the range from 1 to m and will be distinct.

The final line of input contains the log of added ingredients, in order of when they were added. The line begins with an integer k ($1 \leq k \leq 1\,000\,000$) representing the number of ingredients to follow. This is followed by k integers separated by spaces, each representing an ingredient. Note that there will never be a situation in which you're being asked to add an ingredient that is already in the bowl you are filling.

Output

Output a single integer, the number of bowls that were thrown away at the end of the day.

Sample Input 1

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

Sample Output 1

```
1
```

Sample Input 2

```
5 3 3
2 1 3
2 2 4
5 1 2 3 4 5
9 5 3 1 2 4 1 2 3 4
```

Sample Output 2

```
0
```

Problem I
Mix-and-Matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix},\ B = \begin{bmatrix} 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 \\ 15 & 16 & 17 & 18 \end{bmatrix}$$

$$\begin{aligned} AB &= \begin{bmatrix} 1\cdot7+2\cdot11+3\cdot15 & 1\cdot8+2\cdot12+3\cdot16 & 1\cdot9+2\cdot13+3\cdot17 & 1\cdot10+2\cdot14+3\cdot18 \\ 4\cdot7+5\cdot11+6\cdot15 & 4\cdot8+5\cdot12+6\cdot16 & 4\cdot9+5\cdot13+6\cdot17 & 4\cdot10+5\cdot14+6\cdot18 \end{bmatrix} \\ &= \begin{bmatrix} 7+22+45 & 8+24+48 & 9+26+51 & 10+28+54 \\ 28+55+90 & 32+60+96 & 36+65+102 & 40+70+108 \end{bmatrix} \\ &= \begin{bmatrix} 74 & 80 & 86 & 92 \\ 173 & 188 & 203 & 218 \end{bmatrix} \end{aligned}$$

Figure I.1: An example of matrix multiplication.

In order to fairly divide a piece of cake between children, what some parents will enforce is have one of them cut the cake into two slices and have the other choose which slice they want. Assuming both want as much cake as possible, the best strategy in this case is for the first child to cut the cake perfectly evenly.

You and your brother have found yourselves in a similar dilemma. At your recent Christmas party, your math-loving relatives have gifted you two some matrices but were too busy singing along to Mariah Carey to specify who gets which of them. Now, you're in charge of splitting your gifts properly, but the definition of an even split is oddly specific here. Namely, together you have four square matrices and want to split them into groups of two such that the product of the first two equals the product of the last two. You may recall that the definition of the product between an $m \times k$ matrix A and a $k \times n$ matrix B is a new $m \times n$ matrix C such that $\forall i \in \{1, 2, \ldots, m\}, j \in \{1, 2, \ldots, n\}$, we have $C_{ij} = \sum_{t=1}^{k} A_{it} B_{tj}$. As matrix multiplication is not commutative, the order will matter within the two groups.

Since you'll be splitting the gifts, your brother will take care of selecting which ones he gets and which ones you're left with. In fact, he is so hands-off about your method of splitting the matrices that he doesn't even mind if you decide to go with a randomized algorithm that has a nonzero (albeit, very low) probability of being incorrect. When it comes time for him to get his matrices (or for our automated judge to check your answer), though, it better be correct.

If there are multiple ways to split the matrices into 2 groups of two, you can answer with any of them.

Input

The first line of the input contains an integer n ($1 \leq n \leq 1\,000$), which will be used to define the size of the square matrices. Following this are n lines each containing n space-separated integers, containing the elements of the $n \times n$ matrix A. The matrices B, C, and D are given in the same format afterward, in this order, and they each have the same size. Every element M_{ij} of each matrix M ($M \in \{A, B, C, D\}$) in the input is an integer that satisfies $-1\,000 \leq M_{ij} \leq 1\,000$.

Output

If there exists an assignment of (A, B, C, D) to (M_1, M_2, M_3, M_4) that satisfies $M_1 M_2 = M_3 M_4$, then print out M_1 and M_2 on one line and M_3 and M_4 on the second line. Any such assignment will be accepted if it meets these conditions. If there is no such assignment, then print out "`No solution.`" (including the period).

Sample Input 1

```
2
2 1
1 1
1 1
0 1
1 -1
0 1
1 -1
-1 2
```

Sample Output 1

```
A D
B C
```

Sample Input 2

```
3
4 3 1
3 4 1
9 5 1
2 2 8
8 8 4
6 4 8
8 6 2
6 8 2
18 10 2
1 1 4
4 4 2
3 2 4
```

Sample Output 2

```
A B
C D
```

Sample Input 3

```
3
1 0 0
0 1 0
0 0 1
1 2 3
4 5 6
7 8 9
1 2 3
4 5 6
7 8 9
1 0 0
0 1 0
0 0 1
```

Sample Output 3

```
D B
A C
```

Sample Input 4

```
2
3 1
4 1
5 9
2 6
5 3
5 8
9 7
9 3
```

Sample Output 4

```
No solution.
```

This page is intentionally left blank.

Problem J
Pay Your Taxes

Image by Barta IV from Pixabay

In the distant island of Hamt, like in many other countries, people complain that they don't use the math they learned in school for anything. In response to the complaints, the Hamt government decided to change the tax system. Under the new tax system, taxpayers each year receive a formula for the amount that needs to be paid each month.

The formula involves taking the number of the month, raising it to an integer power (possibly multiple times), then multiplying the result with an integer (again possibly multiple times), and finally adding one or more integers.

Mathematically, the formula can be represented in the following format:

$$m^{A_1 \cdot \ldots \cdot A_n} \cdot B_1 \cdot \ldots \cdot B_m + C_1 + \ldots + C_l$$

where m is the month of the year (1 to 12), and A_i, B_j, and C_k are positive integer constants ($1 \le i \le n, 1 \le j \le m, 1 \le k \le l$).

You work for the accounting department of a company. Your boss suspects that the company has been paying its taxes wrong and wants you to recalculate the total amount owed in taxes for some intervals before the government notices.

Input

The input consists of a single test case. The first line contains two integers I and Y, denoting the number of intervals I that need to be calculated and the number of years Y, respectively ($1 \le I \le 500\,000, 1 \le Y \le 50$). The next Y lines contain the formulas for each of the years from 1 to Y, where each formula is no longer than 500 characters. The formulas uses "^" to represent power, "$\star$" to represent multiplication and "+" to represent addition. The next I lines each contain 4 integers y_1, m_1, y_2, m_2 that represent the interval from the month m_1 of the year y_1 to the month m_2 of the year y_2 ($1 \le y_1, y_2 \le Y, 1 \le m_1, m_2 \le 12$).

Output

For each interval output a line with the amount of taxes owed for that interval. You are guaranteed that this amount is less than 2^{63} for each interval in the input.

Sample Input 1	Sample Output 1

`1 1`
`m^2*2+2`
`1 1 1 2`
	`14`

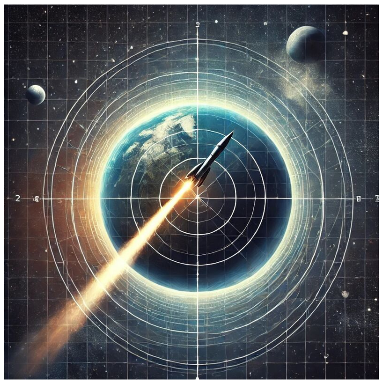
Sample Input 2

```
4 4
m^0+1
m*0+0
m+20
m
4 11 4 12
1 1 4 12
2 10 3 4
1 12 2 1
```

Sample Output 2

```
535
```

Problem K
Meteor Observation


Observing meteors through a fixed telescope

In a distant galaxy, a swarm of meteors is visible from your space station. Mission control has equipped you with a high-powered telescope that can rotate freely but has a limited field of view. Your telescope is fixed at a given position on a suitably aligned, but only two-dimensional (x, y) coordinate plane.

Initially, the telescope points directly east. You can rotate it by any angle, but its field of view is fixed at θ degrees. Your task is to determine the maximum number of meteors that can be observed simultaneously by optimally choosing the telescope's orientation.

A meteor is considered visible if the angle between the telescope's viewing direction and the line from your telescope's position to the meteor is within $\pm\theta/2$ degrees.

Input

The first line contains an integer N ($1 \leq N \leq 200\,000$), the number of meteors. Each of the next N lines contains two integers x_i and y_i ($-10^9 \leq x_i, y_i \leq 10^9$), representing the coordinates of each meteor. No two meteors will be at the same position. The following line contains two integers representing the (x_t, y_t) coordinates of the telescope ($-10^9 \leq x_t, y_t \leq 10^9$), which is different from the position of any meteor. The last line contains a real number θ ($0.001 \leq \theta \leq 360$), representing the telescope's field of view in degrees with up to 3 decimal places.

Output

Print a single integer representing the maximum number of meteors that can be observed at once. You are guaranteed that the answer does not change if θ were increased or decreased by 0.001 degrees.

Sample Input 1	Sample Output 1

10
0 100
100 0
-100 0
0 -100
50 50
-50 -50
75 -75
-75 75
25 -25
-25 25
4 4
45
	3

This page is intentionally left blank.

Problem L
Pairs and Common

Bob starts out with an ordered list a of N positive integers a_i $(1 \leq i \leq N)$. He then performs a total of Q actions that can be of one of two types:

- $1\ x\ y$: Bob adds x copies of value y to the end of the list, whereby y is a prime number.
- $2\ l\ r$: Bob asks you to compute the number of pairs (a_i, a_j) whose greatest common divisor lies in the interval $[l, r]$. That is, to count the number of pairs such that $i < j$ and $l \leq \gcd(a_i, a_j) \leq r$. Since this number can be very large, output its remainder when divided by $10^9 + 7$.

Note that while the original list may contain arbitrary positive integers, type 1 queries will add only prime numbers.

Input

The first line contains one integer N $(1 \leq N \leq 10^6)$. The second line contains N integers a_i $(1 \leq a_i \leq 10^6)$. The third line contains one integer $Q = P + M$ $(1 \leq Q \leq 2 \cdot 10^5)$, where P and M are the numbers of queries of type 1 and 2, respectively.

Each of the next Q lines starts with an integer $t \in \{1, 2\}$:

- If $t = 1$, it is followed by two integers x and y $(1 \leq x, y \leq 10^6)$ where y is prime.
- If $t = 2$, it is followed by two integers l and r $(1 \leq l \leq r \leq 10^6)$.

Output

Output M nonnegative integers on M lines, which are the answers for the M queries of type 2. Remember to output each number modulo $10^9 + 7$.

Sample Input 1

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

Sample Output 1

```
3
7
```

This page is intentionally left blank.