max planck institut informatik

Competitive Programming

Saarland University — Summer Semester 2022

Markus Bläser, Karl Bringmann, Martin Bromberger, Christoph Weidenbach
Julian Baldus, Yasmine Briefs, Simon Schwarz, Dominic Zimmer

Assignments Week 10

Deadline: **July 14, 2022 at 16:00 sharp**

Please submit solutions to the problems in our judge system, available at
`https://compro.mpi-inf.mpg.de/`.
You can find your credentials on your personal status page in our CMS.

Problem	`pathcounter`	`milkman`	`pirates`	`marsmaps`
Points	3	3	3	3
Difficulty				
Time Limit	1s	5s	1s	1s
Memory Limit	2 GB	2 GB	2 GB	2 GB

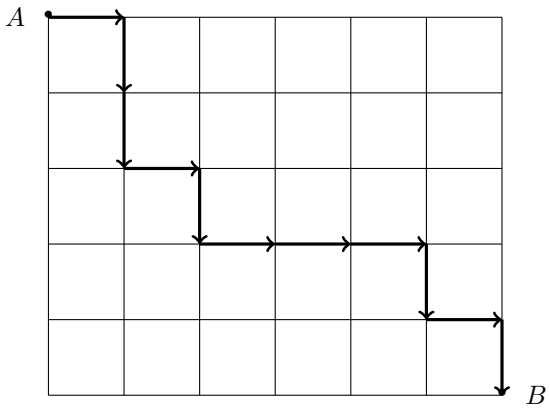
Please note:

- Your solution will be judged immediately after submitting. This may take some time, depending on the current server load.
- You can submit as many times as you want. However, don't abuse the server or try to extract the secret test cases.
- If your solution is **accepted**, you will receive the points specified in the table above.
- If you get **another verdict**, you will receive 0 points.

Path Counter

Problem ID: pathcounter
Time limit: 1 second

In Manhattan, the roads are arranged in a grid-like pattern. We assume that they are directed in the following way: you can only move right or down, never left or up. One possible travel from the top-left corner (point A) to the bottom-right corner (point B) is illustrated in the following picture:



We would like to find out in how many ways we can go from A to B.

Since the answer can be large, output it modulo $998\,244\,353$.

Input

The input consists of two integers N and K in one line, where N is the number of columns (equivalently, this is the number of times you have to move right) and K is the number of rows (equivalently, this is the number of times you have to move down). The numbers are bounded by $1 \leq N \leq 10^9$ and $1 \leq K \leq 10^7$.

Output

Print one integer, the number of paths from the top-left corner to the bottom-right corner, modulo $998\,244\,353$.

Sample Inputs

Note that the first sample corresponds to the grid illustrated above.

Sample Input 1	Sample Output 1
6 5	462

Sample Input 2	Sample Output 2
3 2	10

Sample Input 3	Sample Output 3
2 3	10

Sample Input 4	Sample Output 4
1000000000 10000000	535516646

Milkman

Problem ID: milkman
Time limit: 5 seconds

Linienhausen is a small dullsville in Saarland consisting of N houses arranged in a row, with house numbers from 1 to n. Milky, the sympathic milkman of Linienhausen, delivers fresh milk from the regional cows Gertrude and Paula[1].

Milky always delivers milk for a segment of houses in Linienhausen. Such a delivery of Milky is marked with `? i j` in the input. This means that Milky delivers milk to each house between house number i (inclusive) and house number j (inclusive).

Each house always orders a specific number of units of milk. At the beginning, the house with number i orders v_i units of milk every time Milky visits it. However, sometimes the consumption of milk changes, and some houses want a different amount of milk to be delivered. Such requests are marked with `! i v`, meaning that house number i wants to have v units of milk delivered from now on.

Milky wants his job to be as comfortable as possible. Therefore, for each delivery `? i j`, he wants to choose a size s of the milk containers he is going to use. He will then carry enough containers of that size to fulfill all of the orders. The size s should fulfil the following properties:

- Milky can deliver any amount of whole containers to a house. However, he cannot sell fractions of containers. Therefore, for each customer, the amount of milk ordered should be a multiple of the size of the container. More formally: $\forall i \leq k \leq j : s | v_k$.
- The size s should be as large as possible and it should be a natural number.

Can you help Milky figuring out the optimal size of the containers for each visit?

Input

The first line contains two integers N and Q, the number of houses in Linienhausen and the number of queries, respectively ($1 \leq N \leq 1\,000\,000$, $1 \leq Q \leq 500\,000$). The second line contains the N values v_i, denoting how much milk the i-th house orders initially ($1 \leq v_i \leq 10^9$).

Q lines follow, each of them containing one query. A query has one of the two following forms:

- `! i v`, meaning that the i-th house from now on orders v units of milk ($1 \leq i \leq N$, $1 \leq v \leq 10^9$).
- `? i j`, meaning that you should print the optimal size of the containers needed when delivering to all houses from i to j ($1 \leq i \leq j \leq N$).

Due to relatively large input files, we advise you to use `ios::sync_with_stdio(false)` at the beginning of your code.

Output

For each query starting with `?`, print its result.

[1] Legend also says something about Dieter operating a Disco close to Linienhausen, but this is a completely different story

Sample Input 1	Sample Output 1
5 7 6 9 15 30 6 ? 1 5 ? 4 5 ! 2 12 ! 3 18 ? 1 5 ! 4 5 ? 1 5	3 6 6 1

Dancing Pirates

Problem ID: pirates
Time limit: 1 second

Scary, doing a happy dance

Pirates are a dangerous bunch of people. But whatever you may have heard of the frightening Pirates of the Caribbean, none of them were as dangerous as the crew of Scary McScarface.

One dark night, cruising through the Caribbean in your ship, you accidentally found their secret hideout and stayed to observe their training. That was when their name *Pirates of Dance* finally started to make sense. Their combat was incredibly well structured:

They had agreed to never take more nor less than n steps in their routine, out of which the first - obviously - always had to be a fancy dance move. For every single one of these n steps, there are a number of possible actions, of which they have to choose one. For example their fancy dance move could be a Waltz, a Foxtrot, a back-flip or a head-spin. There might also be a step called *dangerous growl* in which they have to decide whether they shout "Uh!", "Huh!" or "Ha!" menacingly. Since they are aware that it wouldn't be very frightening to first do a Waltz and then scream "Uh!", there are a number of rules for which options in step i could be followed by which options in step $i+1$ such that the routine stays frightening. In addition, you overheard Scary McScarface say that as soon as a routine has been executed once, its capability to intimidate people will vanish. So to stay on the winning side of history, they have to make sure never to repeat a routine.

You know that trying to attack their base will be futile, as long as there are still scary routines left that have not been used before. That's why – looking at your notes – you start to wonder how many attacks this world will have to endure at most before it will be safe to attack the Pirates of Dance.

As the number of attacks may become very large and people may be intimidated by the prospect of that many attacks you decide to compute this number modulo $10^9 + 7$.

Input

Your notes on the rules for the routines that the pirates have agreed upon consist of:

- The number of steps $1 \leq n \leq 314$ on a single line.
- One line containing n space-separated integers $1 \leq a_i \leq 50$ $(1 \leq i \leq n)$ denoting how many options the pirates have agreed upon in step i.

Afterward, there will be $n-1$ blocks. The i-th block will describe which options in step i may be followed by which options in step $i+1$ such that the routine stays frightening.

The i-th block consists of a_i lines of the following form:

- The j-th line will start with an integer $0 \leq m \leq \min(a_{i+1}, 25)$ - the number of possible follow-up options in step $i+1$ to the j-th option in step i
- m distinct space-separated integers $0 \leq b_k < a_{i+1}$ $(k \leq m)$ follow, describing that option b_k in step $i+1$ is safe to be used after option j in step i.

Output

Calculate the number of attacks before the pirates will run out of intimidating routines, assuming that the pirates have not yet performed a routine according to these rules. Print this number modulo $10^9 + 7$.

Sample Input 1

```
2
1 1
1 0
```

Sample Output 1

```
1
```

Sample Input 2

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

Sample Output 2

```
2
```

Sample Input 3

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

Sample Output 3

```
6
```

Mars Maps

Problem ID: marsmaps
Time limit: 1 second

Elon's ambitious plan to settle on mars misses only one important component: a map of the whole planet. Through prior missions, many rovers have collected maps of different rectangular areas of mars. Some of those areas may overlap.

To find out how much effort it still is to map the whole planet, Elon asks you to find out the total area for which maps already exist.

Input

The first line of the input contains N, the number of available rectangular mars maps ($1 \leq N \leq 100\,000$). Each of the next N lines contains one map, represented by four integers x_1, y_1, x_2, y_2 ($0 \leq x_1 < x_2 \leq 100\,000$ and $0 \leq y_1 < y_2 \leq 100\,000$). (x_1, y_1) are the coordinates of the bottom-left corner and (x_2, y_2) are the coordinates of the top-right corner of map rectangle. The sides are parallel to the x- and y-axis of the coordinate system.

Output

Print one integer, the total area of all maps (i.e. the area of their union).

Sample Inputs

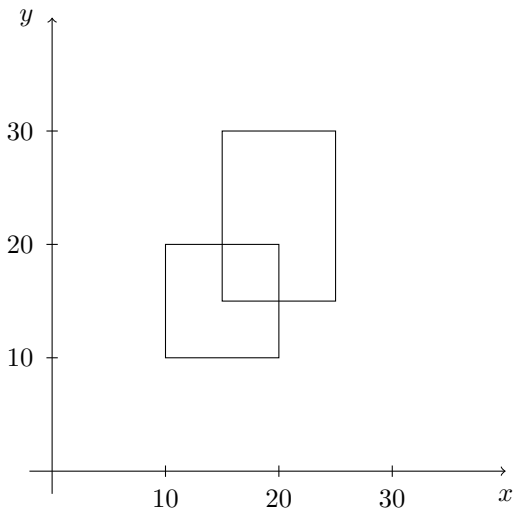


Figure 1: Illustration of the first sample input.

Sample Input 1	Sample Output 1
2 10 10 20 20 15 15 25 30	225

Sample Input 2	Sample Output 2
1 0 0 100000 100000	10000000000

Sample Input 3

```
5
5 5 7 7
0 0 2 10
0 0 10 2
0 8 10 10
8 0 10 10
```

Sample Output 3

```
68
```

Sample Input 4

```
5
1 1 2 2
2 2 4 4
3 3 7 7
4 4 16 16
5 5 17 17
```

Sample Output 4

```
178
```