A. Planning

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Helen works in Metropolis airport. She is responsible for creating a departure schedule. There are n flights that must depart today, the i-th of them is planned to depart at the i-th minute of the day.

Metropolis airport is the main transport hub of Metropolia, so it is difficult to keep the schedule intact. This is exactly the case today: because of technical issues, no flights were able to depart during the first k minutes of the day, so now the new departure schedule must be created.

All n scheduled flights must now depart at different minutes between $(k+1)$-th and $(k+n)$-th, inclusive. However, it's not mandatory for the flights to depart in the same order they were initially scheduled to do so — their order in the new schedule can be different. There is only one restriction: no flight is allowed to depart earlier than it was supposed to depart in the initial schedule.

Helen knows that each minute of delay of the i-th flight costs airport c_i burles. Help her find the order for flights to depart in the new schedule that minimizes the total cost for the airport.

Input

The first line contains two integers n and k $(1 \le k \le n \le 300\,000)$, here n is the number of flights, and k is the number of minutes in the beginning of the day that the flights did not depart.

The second line contains n integers $c_1, c_2, \ldots, c_n$ $(1 \le c_i \le 10^7)$, here c_i is the cost of delaying the i-th flight for one minute.

Output

The first line must contain the minimum possible total cost of delaying the flights.

The second line must contain n different integers $t_1, t_2, \ldots, t_n$ $(k+1 \le t_i \le k+n)$, here t_i is the minute when the i-th flight must depart. If there are several optimal schedules, print any of them.

Example

input

```
5 2
4 2 1 10 2
```

output

```
20
3 6 7 4 5
```

Note

Let us consider sample test. If Helen just moves all flights 2 minutes later preserving the order, the total cost of delaying the flights would be $(3-1)\cdot 4+(4-2)\cdot 2+(5-3)\cdot 1+(6-4)\cdot 10+(7-5)\cdot 2=38$ burles.

However, the better schedule is shown in the sample answer, its cost is $(3-1)\cdot 4+(6-2)\cdot 2+(7-3)\cdot 1+(4-4)\cdot 10+(5-5)\cdot 2=20$ burles.

B. Jury Meeting

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Country of Metropolia is holding Olympiad of Metrpolises soon. It mean that all jury members of the olympiad should meet together in Metropolis (the capital of the country) for the problem preparation process.

There are $n+1$ cities consecutively numbered from 0 to n. City 0 is Metropolis that is the meeting point for all jury members. For each city from 1 to n there is exactly one jury member living there. Olympiad preparation is a long and demanding process that requires k days of work. For all of these k days each of the n jury members should be present in Metropolis to be able to work on problems.

You know the flight schedule in the country (jury members consider themselves important enough to only use flights for transportation). All flights in Metropolia are either going to Metropolis or out of Metropolis. There are no night flights in Metropolia, or in the other words, plane always takes off at the same day it arrives. On his arrival day and departure day jury member is not able to discuss the olympiad. All flights in Megapolia depart and arrive at the same day.

Gather everybody for k days in the capital is a hard objective, doing that while spending the minimum possible money is even harder. Nevertheless, your task is to arrange the cheapest way to bring all of the jury members to Metrpolis, so that they can work together for k days and then send them back to their home cities. Cost of the arrangement is defined as a total cost of tickets for all used flights. It is allowed for jury member to stay in Metropolis for more than k days.

Input

The first line of input contains three integers n, m and k ($1 \le n \le 10^5$, $0 \le m \le 10^5$, $1 \le k \le 10^6$).

The i-th of the following m lines contains the description of the i-th flight defined by four integers d_i, f_i, t_i and c_i ($1 \le d_i \le 10^6$, $0 \le f_i \le n$, $0 \le t_i \le n$, $1 \le c_i \le 10^6$, exactly one of f_i and t_i equals zero), the day of departure (and arrival), the departure city, the arrival city and the ticket cost.

Output

Output the only integer that is the minimum cost of gathering all jury members in city 0 for k days and then sending them back to their home cities.

If it is impossible to gather everybody in Metropolis for k days and then send them back to their home cities, output "`-1`" (without the quotes).

Examples

input

```
2 6 5
1 1 0 5000
3 2 0 5500
2 2 0 6000
15 0 2 9000
9 0 1 7000
8 0 2 6500
```

output

```
24500
```

input

```
2 4 5
1 2 0 5000
2 1 0 4500
2 1 0 3000
8 0 1 6000
```

output

```
-1
```

Note

The optimal way to gather everybody in Metropolis in the first sample test is to use flights that take place on days 1, 2, 8 and 9. The only alternative option is to send jury member from second city back home on day 15, that would cost 2500 more.

In the second sample it is impossible to send jury member from city 2 back home from Metropolis.

C. Boredom

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Ilya is sitting in a waiting area of Metropolis airport and is bored of looking at time table that shows again and again that his plane is delayed. So he took out a sheet of paper and decided to solve some problems.

First Ilya has drawn a grid of size $n \times n$ and marked n squares on it, such that no two marked squares share the same row or the same column. He calls a rectangle on a grid with sides parallel to grid sides *beautiful* if exactly two of its corner squares are marked. There are exactly $n \cdot (n - 1) / 2$ beautiful rectangles.

Ilya has chosen q query rectangles on a grid with sides parallel to grid sides (not necessarily beautiful ones), and for each of those rectangles he wants to find its *beauty degree*. Beauty degree of a rectangle is the number of beautiful rectangles that share at least one square with the given one.

Now Ilya thinks that he might not have enough time to solve the problem till the departure of his flight. You are given the description of marked cells and the query rectangles, help Ilya find the beauty degree of each of the query rectangles.

Input

The first line of input contains two integers n and q ($2 \le n \le 200\ 000$, $1 \le q \le 200\ 000$) — the size of the grid and the number of query rectangles.

The second line contains n integers $p_1, p_2, \ldots, p_n$, separated by spaces ($1 \le p_i \le n$, all p_i are different), they specify grid squares marked by Ilya: in column i he has marked a square at row p_i, rows are numbered from 1 to n, bottom to top, columns are numbered from 1 to n, left to right.

The following q lines describe query rectangles. Each rectangle is described by four integers: l, d, r, u ($1 \le l \le r \le n$, $1 \le d \le u \le n$), here l and r are the leftmost and the rightmost columns of the rectangle, d and u the bottommost and the topmost rows of the rectangle.

Output

For each query rectangle output its beauty degree on a separate line.

Examples

input

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

output

```
1
1
1
```

input

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

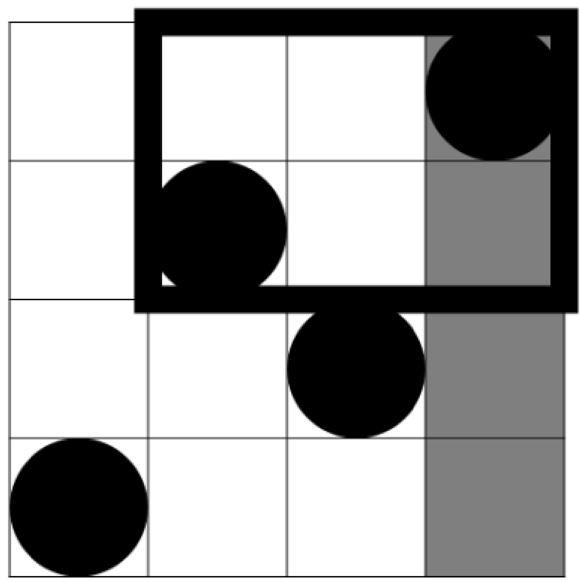
output

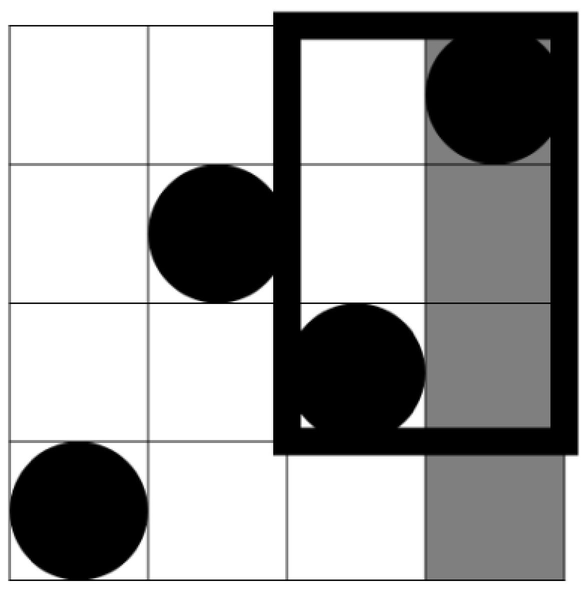
```
3
5
```

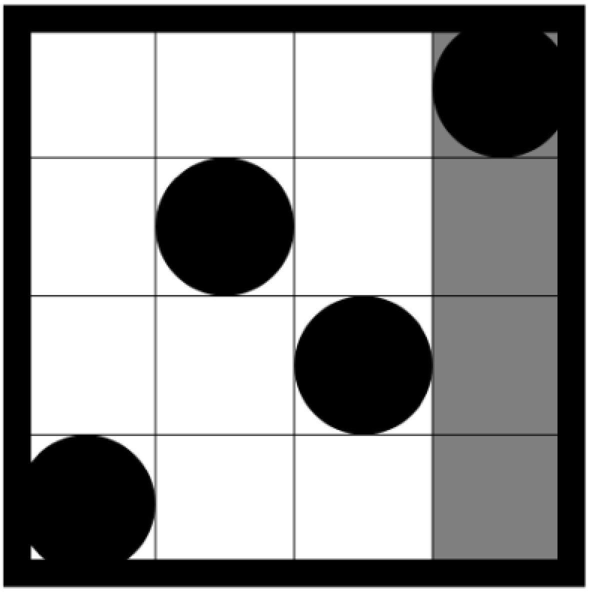
Note

The first sample test has one beautiful rectangle that occupies the whole grid, therefore the answer to any query is 1.

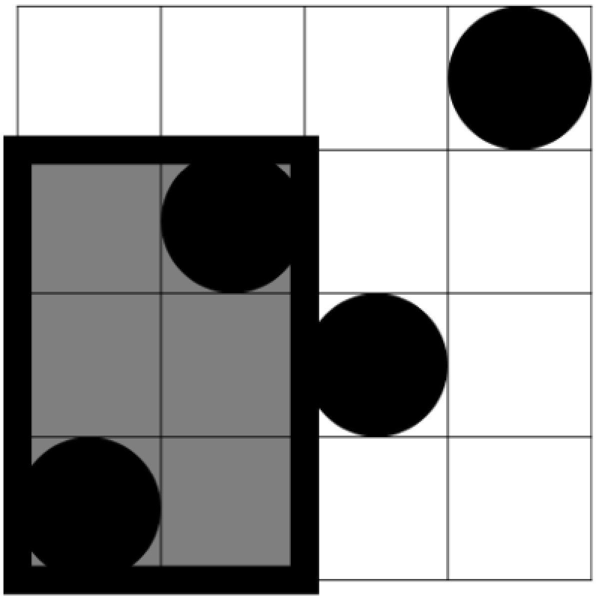
In the second sample test the first query rectangle intersects 3 beautiful rectangles, as shown on the picture below:

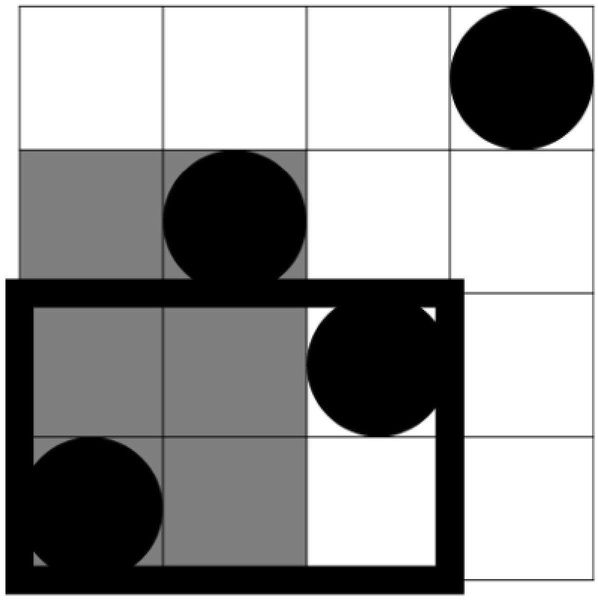


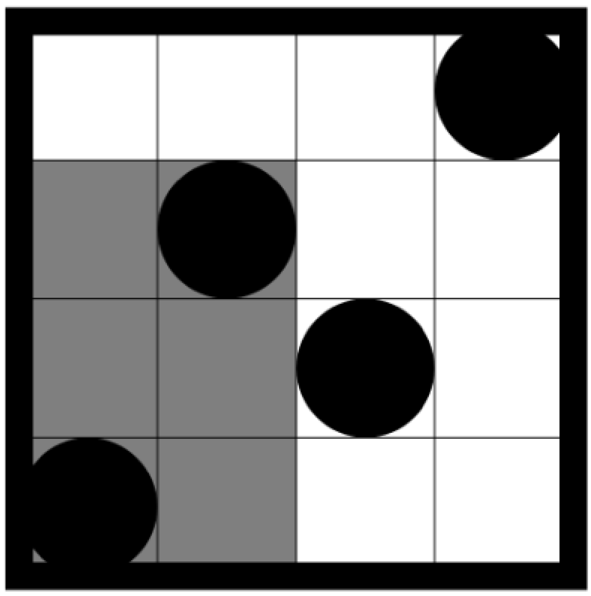


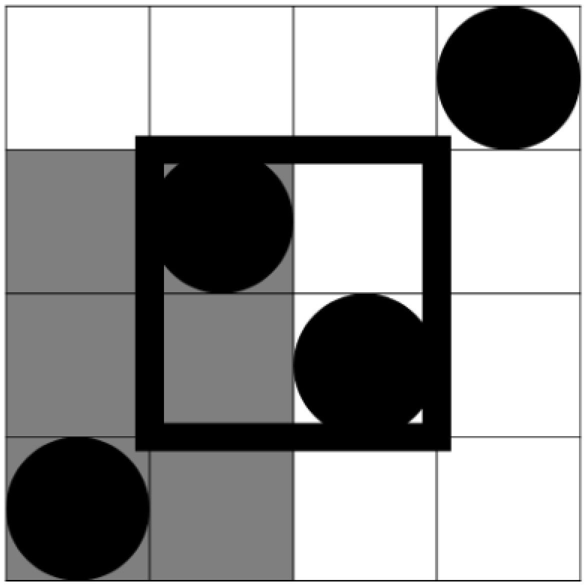


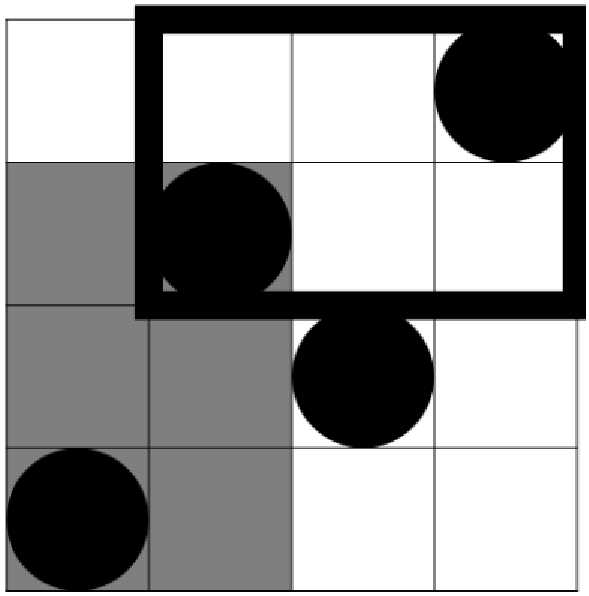
There are 5 beautiful rectangles that intersect the second query rectangle, as shown on the following picture:











D. Michael and Charging Stations

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Michael has just bought a new electric car for moving across city. Michael does not like to overwork, so each day he drives to only one of two his jobs.

Michael's day starts from charging his electric car for getting to the work and back. He spends 1000 burles on charge if he goes to the first job, and 2000 burles if he goes to the second job.

On a charging station he uses there is a loyalty program that involves bonus cards. Bonus card may have some non-negative amount of bonus burles. Each time customer is going to buy something for the price of x burles, he is allowed to pay an amount of y ($0 \leq y \leq x$) burles that does not exceed the bonus card balance with bonus burles. In this case he pays $x - y$ burles with cash, and the balance on the bonus card is decreased by y bonus burles.

If customer pays whole price with cash (i.e., $y = 0$) then 10% of price is returned back to the bonus card. This means that bonus card balance increases by $\frac{x}{10}$ bonus burles. Initially the bonus card balance is equal to 0 bonus burles.

Michael has planned next n days and he knows how much does the charge cost on each of those days. Help Michael determine the minimum amount of burles in cash he has to spend with optimal use of bonus card. Assume that Michael is able to cover any part of the price with cash in any day. It is not necessary to spend all bonus burles at the end of the given period.

Input

The first line of input contains a single integer n ($1 \leq n \leq 300\,000$), the number of days Michael has planned.

Next line contains n integers $a_1, a_2, \ldots, a_n$ ($a_i = 1000$ or $a_i = 2000$) with a_i denoting the charging cost at the day i.

Output

Output the minimum amount of burles Michael has to spend.

Examples

input

```
3
1000 2000 1000
```

output

```
3700
```

input

```
6
2000 2000 2000 2000 2000 1000
```

output

```
10000
```

Note

In the first sample case the most optimal way for Michael is to pay for the first two days spending 3000 burles and get 300 bonus burles as return. After that he is able to pay only 700 burles for the third days, covering the rest of the price with bonus burles.

In the second sample case the most optimal way for Michael is to pay the whole price for the first five days, getting 1000 bonus burles as return and being able to use them on the last day without paying anything in cash.

E. Lada Malina

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

After long-term research and lots of experiments leading Megapolian automobile manufacturer «AutoVoz» released a brand new car model named «Lada Malina». One of the most impressive features of «Lada Malina» is its highly efficient environment-friendly engines.

Consider car as a point in Oxy plane. Car is equipped with k engines numbered from 1 to k. Each engine is defined by its velocity vector whose coordinates are (vx_i, vy_i) measured in distance units per day. An engine may be turned on at any level w_i, that is a real number between -1 and $+1$ (inclusive) that result in a term of $(w_i \cdot vx_i, w_i \cdot vy_i)$ in the final car velocity. Namely, the final car velocity is equal to

$$(w_1 \cdot vx_1 + w_2 \cdot vx_2 + \ldots + w_k \cdot vx_k, \quad w_1 \cdot vy_1 + w_2 \cdot vy_2 + \ldots + w_k \cdot vy_k)$$

Formally, if car moves with constant values of w_i during the whole day then its x-coordinate will change by the first component of an expression above, and its y-coordinate will change by the second component of an expression above. For example, if all w_i are equal to zero, the car won't move, and if all w_i are equal to zero except $w_1 = 1$, then car will move with the velocity of the first engine.

There are n factories in Megapolia, i-th of them is located in (fx_i, fy_i). On the i-th factory there are a_i cars «Lada Malina» that are ready for operation.

As an attempt to increase sales of a new car, «AutoVoz» is going to hold an international exposition of cars. There are q options of exposition location and time, in the i-th of them exposition will happen in a point with coordinates (px_i, py_i) in t_i days.

Of course, at the «AutoVoz» is going to bring as much new cars from factories as possible to the place of exposition. Cars are going to be moved by enabling their engines on some certain levels, such that at the beginning of an exposition car gets exactly to the exposition location.

However, for some of the options it may be impossible to bring cars from some of the factories to the exposition location by the moment of an exposition. Your task is to determine for each of the options of exposition location and time how many cars will be able to get there by the beginning of an exposition.

Input

The first line of input contains three integers k, n, q ($2 \le k \le 10$, $1 \le n \le 10^5$, $1 \le q \le 10^5$), the number of engines of «Lada Malina», number of factories producing «Lada Malina» and number of options of an exposition time and location respectively.

The following k lines contain the descriptions of «Lada Malina» engines. The i-th of them contains two integers vx_i, vy_i ($-1000 \le vx_i, vy_i \le 1000$) defining the velocity vector of the i-th engine. Velocity vector can't be zero, i.e. at least one of vx_i and vy_i is not equal to zero. It is guaranteed that no two velosity vectors are collinear (parallel).

Next n lines contain the descriptions of factories. The i-th of them contains two integers fx_i, fy_i, a_i ($-10^9 \le fx_i, fy_i \le 10^9$, $1 \le a_i \le 10^9$) defining the coordinates of the i-th factory location and the number of cars that are located there.

The following q lines contain the descriptions of the car exposition. The i-th of them contains three integers px_i, py_i, t_i ($-10^9 \le px_i, py_i \le 10^9$, $1 \le t_i \le 10^5$) defining the coordinates of the exposition location and the number of days till the exposition start in the i-th option.

Output

For each possible option of the exposition output the number of cars that will be able to get to the exposition location by the moment of its beginning.

Examples

input

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

output

```
3
```

input

```
3 4 3
2 0
-1 1
-1 -2
-3 0 6
1 -2 1
-3 -7 3
3 2 2
-1 -4 1
0 4 2
6 0 1
```

output

```
4
9
0
```

Note

Images describing sample tests are given below. Exposition options are denoted with crosses, factories are denoted with points. Each factory is labeled with a number of cars that it has.

First sample test explanation:

- Car from the first factory is not able to get to the exposition location in time.
- Car from the second factory can get to the exposition in time if we set $w_1 = 0$, $w_2 = 1$.
- Car from the third factory can get to the exposition in time if we set $w_1 = \frac{1}{4}$, $w_2 = \frac{3}{4}$.
- Car from the fourth factory can get to the exposition in time if we set $w_1 = 1$, $w_2 = 0$.