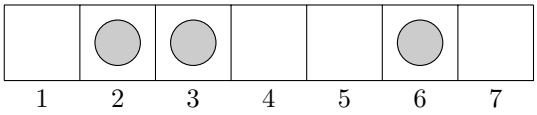
Task: GRA
Game

Brazilian ICPC Summer School 2018, Day 3. Available memory: 64 MB. *31.01.2018*

Let us consider a game on a rectangular board $m \times 1$ consisting of m elementary squares numbered successively from 1 to m. There are n pawns on the board, each on a distinct square. None of them occupies the square with number m. Each single move in the is the following action: the moving player picks a pawn from any occupied square chosen at will and places it on the first unoccupied square with a larger number. The two players make moves in turn. The one who puts a pawn on the last square, i.e. the square with a number m, wins.



In the case presented in the figure ($m = 7$), a player is allowed to move a pawn from square number 2 to 4, from square 3 to 4 or from square 6 to 7. The latter ends the game.

We say a player's move is winning if after making it he can win the game, no matter what moves his opponent makes.

Write a programme that: reads the size of a board and the initial setup of pawns from the standard input, and determines the number of distinct winning moves the starting player may choose in the given initial situation.

Input

The first line of the input contains two integers m and n ($2 \leq m \leq 10^9$, $1 \leq n \leq 10^6$, $n < m$) separated by a single space. The second line contains n increasing numbers – these are the numbers of squares the pawns are set on. Numbers in the line are separated by single spaces.

Output

The first and only output line should contain the number of distinct winning moves possible for the starting player in the given initial situation.

Example

For the input data:

```
5 2
1 3
```

the correct result is:

```
1
```

and for the input data:

```
5 2
2 3
```

the correct answer is:

```
0
```

Task: KLU
Club Members

Brazilian ICPC Summer School 2018, Day 3. Available memory: 256 MB. *31.01.2018*

Byteotian Discussion Club is most extraordinary in its every aspect. Each of its 2^n members has filled out a questionnaire containing n fundamental Yes or No questions. Each member's answers can of course be encoded as a sequence of n bits, which yields an integer in the range from 0 to $2^n - 1$. We are going to ignore the specifics of questions formulations or the mapping of Yes and No answers to 0 and 1. Instead, we list some extraordinary facts about the club members below.

No two club members have given the same answers, i.e., each number in the aforementioned range is present. Moreover, exactly 2^{n-1} of the club members are men and the remaining 2^{n-1} members are women. If this were not extraordinary enough, they form in fact 2^{n-1} couples. During club sessions, the members sit at a *round* table. We would like to sit them so that every member sits between to their partner and a *nearly agreeing* member, i.e., one who answered only a single question differently.

Input

In the first line of the standard input, there is an integer n ($2 \leq n \leq 19$) specifying the number of fundamental questions. The following 2^{n-1} lines describe the member couples: the i-th such line contains two integers a_i, b_i ($0 \leq a_i, b_i \leq 2^n - 1$), separated by a single space, which indicate that the club members whose questionnaire answers are encoded by a_i and b_i are a couple. Each of the 2^n numbers representing answers is going to appear on input exactly once.

Output

A single line should be printed to the standard output, containing the word `NIE` (Polish for *no*) if no placement of club members satisfies aforementioned requirements, and a valid placement otherwise; the latter should be a sequence of 2^n integers (encoding the answers of successive members along the table), separated by single spaces.

If there is more than one correct answer, print any of those.

Example

For the input data:

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

the correct result is:

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

Task: KOZ
The Goat

Brazilian ICPC Summer School 2018, Day 3. Available memory: 256 MB. *31.01.2018*

Farmer Byteman took his sledgehammer and drove n stakes into the ground of an infinite pasture. Through the next k days each morning farmer Byteman takes his goat out to the pasture and ties it to a randomly selected stake with a cord of length l. During the day the goat eats all the grass in its reach. To the chargin of the goat the grass does not grow back. Moreover, it can happen that the scatterbrained farmer will tie the goat more than once to the same stake.

What is the expected value of the area of the pasture on which the grass will be eaten after k days?

Input

In the first line of the standard input contains three integers n, k and l ($1 \leq n, k, l \leq 1\,000$) denoting respectively the number of stakes, the number of days, and the length of the cord. Each of the next n lines contains the coordinates of one stake in the form of a pair of integers x_i, y_i ($-1\,000 \leq x_i, y_i \leq 1\,000$). No two stakes are driven into the ground in the same place.

Output

In the only line of the standard output a single real number should be written. The number should be the expected value of the area of the part of the pasture from which the goat will have eaten the grass during k days. The answer will be accepted if it will differ from the correct answer by no more than 10^{-6}. No more than 20 digits should be given after the decimal point.

Example

For the input data:

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

the correct result is:

```
4.098204131080311
```

Explanation of the example: If the goat will be tied to the same stake during both days the area of the eaten grass will be equal to π. On the other hand if it will be tied to two different stakes, the area will be equal to $\frac{4}{3}\pi + \frac{\sqrt{3}}{2}$. Thus the answer is $\frac{7}{6}\pi + \frac{\sqrt{3}}{4}$.

Task: KUR
Couriers

Brazilian ICPC Summer School 2018, Day 3. Available memory: 128 MB. *31.01.2018*

Byteasar works for the BAJ company, which sells computer games. The BAJ company cooperates with many courier companies that deliver the games sold by the BAJ company to its customers. Byteasar is inspecting the cooperation of the BAJ company with the couriers. He has a log of successive packages with the courier company that made the delivery specified for each package. He wants to make sure that no courier company had an unfair advantage over the others.

If a given courier company delivered more than half of all packages sent in some period of time, we say that it *dominated* in that period. Byteasar wants to find out which courier companies dominated in certain periods of time, if any.

Help Byteasar out! Write a program that determines a dominating courier company or that there was none.

Input

The first line of the standard input contains two integers, n and m ($1 \leq n, m \leq 500\,000$), separated by a single space, that are the number of packages shipped by the BAJ company and the number of time periods for which the dominating courier is to be determined, respectively. The courier companies are numbered from to (at most) n.

The second line of input contains n integers, $p_1, p_2, \ldots, p_n$ ($1 \leq p_i \leq n$), separated by single spaces; p_i is the number of the courier company that delivered the i-th package (in shipment chronology).

The m lines that follow specify the time period queries, one per line. Each query is specified by two integers, a and b ($1 \leq a \leq b \leq n$), separated by a single space. These mean that the courier company dominating in the period between the shipments of the a-th and the b-th package, including those, is to be determined.

Output

The answers to successive queries should be printed to the standard output, one per line. (Thus a total of m lines should be printed.) Each line should hold a single integer: the number of the courier company that dominated in the corresponding time period, or 0 if there was no such company.

Example

For the input data:

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

the correct result is:

```
1
0
3
0
4
```

Task: MEB
Furniture

Brazilian ICPC Summer School 2018, Day 3. Available memory: 128 MB. *31.01.2018*

Byteasar wants to arrange his apartment. He went to a BITKEA store and bought n types of furniture; exactly c_i units of furniture of type i.

He needs a_i minutes to assemble the first unit of furniture of type i. To assemble every other unit of type i he needs d_i minutes less than to assemble previous item of the same type (therefore $a_i - d_i$ minutes for the second unit, $a_i - 2 \cdot d_i$ minutes for the third unit, and so on).

Byteasar decided that today he will assemble some number of units. He would like to know for each value m_i what is the minimal number of minutes he needs to assemble some m_i of units.

Input

In the first line of the input there are two integers n and k ($1 \leq n, k \leq 500$) denoting the number of types of furniture and the number of values m_i.

In the i-th of subsequent n lines there are three integers a_i, d_i, c_i ($1 \leq a_i, d_i, c_i \leq 10^9, a_i > (c_i - 1) \cdot d_i$), which describe the i-th type of furniture.

In the i-th of subsequent k lines there is an integer m_i ($1 \leq m_i \leq 20\,000$).

Output

You need to write k lines to the output; in the i-th line you need to write an integer denoting the minimal number of minutes that is needed to assemble some m_i units of furniture. You can assume that it will be possible for all values m_i from the input.

Example

For the input data:

```
3 6
20 3 6
25 20 2
19 1 19
1
2
3
4
5
6
```

the correct result is:

```
19
30
49
62
70
75
```

Task: MET
Meteors

Brazilian ICPC Summer School 2018, Day 3. Available memory: 64 MB. *31.01.2018*

Byteotian Interstellar Union (BIU) has recently discovered a new planet in a nearby galaxy. The planet is unsuitable for colonisation due to strange meteor showers, which on the other hand make it an exceptionally interesting object of study.

The member states of BIU have already placed space stations close to the planet's orbit. The stations' goal is to take samples of the rocks flying by. The BIU Commission has partitioned the orbit into m sectors, numbered from 1 to m, where the sectors 1 and m are adjacent. In each sector there is a single space station, belonging to one of the n member states.

Each state has declared a number of meteor samples it intends to gather before the mission ends. Your task is to determine, for each state, when it can stop taking samples, based on the meter shower predictions for the years to come.

Input

The first line of the input gives two integers, n and m ($1 \leq n, m \leq 300\,000$), separated by a single space, that denote, respectively, the number of BIU member states and the number of sectors the orbit has been partitioned into.

In the second line there are m integers $o_1, o_2, \ldots, o_m$ ($1 \leq o_i \leq n$), separated by single spaces, that denote the states owning stations in successive sectors.

In the third line there are n integers $p_1, p_2, \ldots, p_n$ ($1 \leq p_i \leq 10^9$), separated by single spaces, that denote the numbers of meteor samples that the successive states intend to gather.

In the fourth line there is a single integer k ($1 \leq k \leq 300\,000$) that denotes the number of meteor showers predictions. The following k lines specify the (predicted) meteor showers chronologically. The i-th of these lines holds three integers l_i, r_i, a_i (separated by single spaces), which denote that a meteor shower is expected in sectors $l_i, l_{i+1}, \ldots, r_i$ (if $l_i \leq r_i$) or sectors $l_i, l_{i+1}, \ldots, m, 1, \ldots, r_i$ (if $l_i > r_i$), which should provide each station in those sectors with a_i meteor samples ($1 \leq a_i \leq 10^9$).

Output

Your program should print n lines on the output. The i-th of them should contain a single integer w_i, denoting the number of shower after which the stations belonging to the i-th state are expected to gather at least p_i samples, or the word `NIE` (Polish for *no*) if that state is not expected to gather enough samples in the foreseeable future.

Example

For the input data:

```
3 5
1 3 2 1 3
10 5 7
3
4 2 4
1 3 1
3 5 2
```

the correct result is:

```
3
NIE
1
```

Task: NAJ
The Invasion

Brazilian ICPC Summer School 2018, Day 3. Available memory: 64 MB. *31.01.2018*

And so it has come – the Triangles have invaded Byteotia! Byteotia lies on an island, occupying its entire surface. The shape of the island is a convex polygon (i.e. a polygon whose each inner angle is smaller than 180°). A certain number of software factories are located in Byteotia, each of which generates constant gains or losses.

The Triangles have decided to occupy such a part of Byteotia which:

- is a triangle-shaped area, the vertices of which are three different vertices of the polygon-island,
- brings the largest income i.e. the sum of all gains and losses generated by factories within the occupied area is maximal.

We assume that a factory located on the border or in the vertex of occupied area belongs to that area. A territory which contains no factory brings, obviously, a zero income.

Byteasar, the King of Byteotia, is concerned by the amount of losses the Triangles' invasion could generate. Help him by writing a program which shall calculate the sum of gains and losses generated by factories which the Triangles wish to capture.

Input

The first line of the input contains a single integer n ($3 \leq n \leq 600$), denoting the number of vertices of the polygon-island. The following n lines of the input contain two integers each x_j and y_j ($-10\,000 \leq x_j, y_j \leq 10\,000$), separated by a single space, denoting the coordinates of consecutive vertices of the island, in a clockwise order.

The next line contains a single integer m ($1 \leq m \leq 10\,000$), denoting the total number of factories. In each of the following m lines there are three integers x'_i, y'_i and w_i ($-10\,000 \leq x'_i, y'_i \leq 10\,000$, $-100\,000 \leq w_i \leq 100\,000$), separated by single spaces, denoting: the coordinates of the i-th factory and the gain (for $w_i \geq 0$) or loss (for $w_i < 0$) this factory generates, respectively. Each factory is situated on the polygon-island i.e. within or on the border of it. Distinct factories may be located in the same place i.e. have the same coordinates.

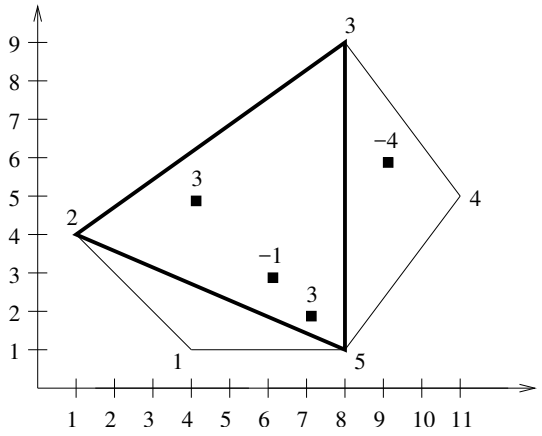
Output

The first and only line of the output should contain a single integer denoting the maximal value of sum of all gains and losses generated by factories within a triangle whose vertices are three different vertices of the polygon-island. Notice that it may happen that the outcome is a negative integer.

Example

For the input data:

```
5
4 1
1 4
8 9
11 5
8 1
4
7 2 3
6 3 -1
4 5 3
9 6 -4
```



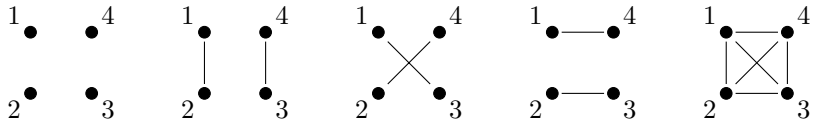
the correct result is:

```
5
```

Task: POW
Return of the Cliquers

Brazilian ICPC Summer School 2018, Day 3. Available memory: 64 MB. *31.01.2018*

An undirected graph with n vertices is called **a symmetric labeled cliquer** if each connected component of the graph contains the same number of vertices and is a clique, and the vertices of the graph are numbered with numbers from the set $\{1, \ldots, n\}$. Maurycy has drawn all symmetric labeled cliquers on a piece of paper and is going to assess beauty of each of them with a number from the set $\{1, \ldots, m\}$ (in particular, different cliquers may be assigned equal grades). In how many ways can he do this? The result should be computed modulo $10^9 - 401$. The figure below depicts all symmetric labeled cliquers for $n = 4$.



Input

The only line of the standard input contains two integers n and m ($1 \leq n, m \leq 2 \cdot 10^9$), separated by a single space and denoting the number of vertices of each symmetric labeled cliquer and the number of grades respectively.

Output

The only line of the standard output should contain the number of possible sets of grades modulo $10^9 - 401$.

Example

For the input data:	the correct result is:
`4 2`	`32`

Task: ROZ
Variety

Brazilian ICPC Summer School 2018, Day 3. Available memory: 512 MB. *31.01.2018*

You are given a two-dimensional array of integers A which has m rows and n columns. Each subarray of A of size $k \times k$ is called a *k-fragment*.

By *variety* of a k-fragment we mean a number of its different elements. Your task is to calculate a maximal variety among all k-fragments of array A and the sum of varieties of all k-fragments.

Input

The first line of the input consists of three integers m, n, k ($1 \leq n, m, k \leq 3000$, $k \leq \min(m, n)$) specifying the size of the array and fragments.

Each subsequent m rows consists of n integers – they specify subsequent rows of array A. The numbers are from the range $[1, 100\,000]$.

Output

You have to write to the output two numbers separated by a single space: the maximal variety of a k-fragment and the sum of varieties of all k-fragments.

Example

For the input data:	the correct result is:

`3 5 2`
`1 5 3 3 3`
`4 1 3 3 4`
`4 2 4 4 3`
	`4 20`

Explanation to the example: Subsequent 2-fragments (from left to right) in the top row have varieties 3, 3, 1 and 2, and 2-fragments below have varieties 3, 4, 2 and 2.

Task: ZAP
Panini 2

Brazilian ICPC Summer School 2018, Day 3. Available memory: 128 MB. *31.01.2018*

Byteasar runs a world-famous panini store, known for its products quality and unique flavor. Over the years, he attained a stable market position and a group of patrons.

Everyday, n customers come to his store, the i-th one arriving at time t_i. (Byteasar shows up at his store at time 0). Every customer orders a single panino. Byteasar can grill at most one panino at a time, each grilling process takes exactly the same units of time, and the grilling cannot be paused, stopped, or interfered with in any way.

The Byteasar's grilling machine is old, so it has to be replaced by a new one. The crucial parameter of a machine is the duration of grilling one panino. Of course, Byteasar would like to have the fastest machine available, but fast machines the expensive, so he has to make some trade off.

Byteasar would like to minimize the total service / waiting time of his customers. (Byteasar knows his customers so well that he knows their arrival times and favorite panini, so he may start grilling them in advance, before a customer arrives. However, no customer's order may be ready before they arrive, as no one likes a cold panino!) He decided to calculate this time for different machines and then decide which one to buy.

There are m machines available, the i-th machine grills one panino in time d_i. Help Byteasar and calculate for each machine the expected waiting time.

Input

The first line of the input contains two integers n and m ($1 \leq n, m \leq 200\,000$) specifying the number of clients and machines, respectively. In the second line, there is a sequence of n integers $t_1, t_2, \ldots, t_n$ ($0 \leq t_1 \leq t_2 \leq \ldots \leq t_n \leq 10^{12}$); the number t_i is the arrival time of the i-th customer. It is possible that more than one client arrives at the same time.

In the third line, there is a sequence of m integers $d_1, d_2, \ldots, d_m$ ($1 \leq d_i \leq 10^6$); the number d_i is the duration of grilling process for the i-th machine.

Output

You have to output exactly m lines; the i-th line should contain one integer – the total service / waiting time of the customers in the optimum grilling schedule for i-th machine.

Example

For the input data:	the correct result is:

`4 3`
`3 10 11 23`
`4 2 5`
	`4`
	`1`
	`6`