Problem A. Memory Copy

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

One autumn rainy day, programmer Alexey was trying to pass programming exam in MSU for the fifth time. His teacher Andrew-Six-Meters was bored, so he just asked Alexey to optimize a very simple program while he go buy some beer.

This program was making some operations with array `X` of n integers. Andrew said that it only copies segments of this array and sometimes fills some segment of the array with consecutive integers. Here are the three procedures Andrew is applying to the array, written in C-style pseudocode. Assume that `X` is an 1-indexed global integer array of size n and `last` is a global integer variable initially equal to 0.

```
void forwardCopy(int a, int b, int l)
{
  for (int i = 0; i < l; i++)
  {
    X[b] = X[a];
    a++;
    b++;
  }
}

void backwardCopy(int a, int b, int l)
{
  for (int i = 0; i < l; i++)
  {
    X[a] = X[b];
    a--;
    b--;
  }
}

void fill(int a, int b)
{
  for (int i = a; i <= b; i++)
  {
    last++;
    X[i] = last;
  }
}
```

Andrew's program starts with the call "`fill(1, n)`", and after that, it applies the above procedures. The order of applying procedures and their arguments are given. Your task is to find the contents of array `X` after all operations. Andrew guarantees that variable `last` will not exceed $5n$ at any moment of execution and that each procedure call is valid (i. e. all writes to the array will be within array bounds).

Input

The first line of input contains two integers n and m ($1 \le n, m \le 200\,000$) — the size of array `X` and the number of operations.

Next m lines contain description of operations in the following form:

- `f a b l` — call of `forwardCopy(a, b, l)`
- `b a b l` — call of `backwardCopy(a, b, l)`
- `x a b` — call of `fill(a, b)`

Output

On the first line of output, print the contents of array `X` after applying all the given operations.

Example

standard input	standard output
10 3 b 2 10 2 x 7 9 f 2 5 4	9 10 3 4 10 3 4 10 13 10

Problem B. Polynomials

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Consider ring $\mathbb{Z}[x]$ of polynomials of one variable x with integer coefficients. A polynomial $p(x) \in \mathbb{Z}[x]$ of degree $n \geq 0$ is equal to $a_0 + a_1 x + \ldots + a_n x^n$ for some integers a_0, a_1, …, a_n such that $a_n \neq 0$. A polynomial $q(x) \in \mathbb{Z}[x]$ is divisible by $p(x) \in \mathbb{Z}[x]$ if there is a polynomial $r(x) \in \mathbb{Q}[x]$ with rational coefficients such that $q(x) = r(x) \cdot p(x)$.

Given a polynomial $p(x) \in \mathbb{Z}[x]$ and an integer $m \geq 2$, find a non-zero polynomial $q(x) \in \mathbb{Z}[x]$ such that the following conditions are satisfied:

- $q(x)$ is divisible by $p(x)$,
- $q(x)$ has the form $b_0 + b_1 x^m + \ldots + b_k x^{km}$ for some integers b_0, b_1, …, b_k such that $b_k \neq 0$.

Input

The first line of input contains two integers n and m ($0 \leq n \leq 2000$, $2 \leq m \leq 5$). The second line contains $n+1$ integers: coefficients a_0, a_1, …, a_n. All coefficients do not exceed 10 by absolute value. It is guaranteed that $a_n \neq 0$.

Output

On the first line of output, print an integer number k ($0 \leq k \leq 4000$) such that degree of $q(x)$ is $m \cdot k$. On the second line, print $k+1$ integers: coefficients b_0, b_1, … b_k. All these coefficients must not exceed 10^{18} by absolute value, the last coefficient b_k must be non-zero. It is guaranteed that there exists at least one answer which satisfies these constraints. If there are several solutions, you may output any one of them.

Examples

standard input	standard output
2 3 2 2 2	1 -1 1
1 4 1 1	1 1 -1
4 2 3 0 1 0 5	2 3 1 5

Problem C. Palindromes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Given is a string S. Find the number of triples of integers (l, r, q) such that $1 \le l \le r \le |S|$, $0 \le q \le r - l$ and the string $S_{l+q}S_{l+q+1} \dots S_r S_l S_{l+1} \dots S_{l+q-1}$ (i. e. the substring $S_l S_{l+1} \dots S_r$ shifted cyclically by q positions to the left) is a palindrome.

Input

The only line of input contains the string S ($1 \le |S| \le 200\,000$). This string consists of lowercase English letters.

Output

On the first line of output, print one integer: the number of such triples.

Examples

standard input	standard output
aba	4
aaa	10

Problem D. TV

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

In the year 3000, TV is still very popular but its technical aspects have changed. It can be briefly described as follows. There are N towers, each tower can be represented as point on the plane with coordinates (x_i, y_i). A point of the plane is covered by TV network if and only if there are three towers such that this point is inside the triangle formed by these towers.

Due to the fact that the towers were commissioned in the course of centuries, the versions of software used on them vary significantly. It is known which version is used on each tower: i-th tower uses version number v_i. Quite naturally, the network began to give failures due to the large differences in the versions.

Experts claim that, if software version numbers on every two towers differed by not more than M, there would be no failures. The owners of the network have reached the only logical decision to destroy some towers so that the network will satisfy the above requirement. Additionally, they decided to choose one of the oldest towers (i. e. a tower with the minimal number of software version) among the towers which were not destroyed and build a museum next to that tower. If there are several such towers, any of them can be chosen.

Now the owners wonder, for each tower i, what will be the maximal possible area of network coverage if the above plan is put into practice and results in building a museum next to tower i.

Input

The first line of input contains two integers N and M ($1 \le N \le 50\,000$, $0 \le M \le 10^9$). The next N lines contain the descriptions of towers; i-th of these lines contains three integers x_i, y_i and v_i: coordinates and software version number of i-th tower. It is guaranteed that $-5000 \le x_i, y_i \le 5000$, $1 \le v_i \le 10^9$ and there are no two towers located in one point.

Output

On the first N lines of output, print N real numbers, one per line; i-th number must be the maximum area of network coverage given the fact that the museum is built next to i-th tower. Each number must be output on a separate line. Absolute or relative error must be at most 10^{-4}.

Examples

standard input	standard output
5 2 0 0 1 2 0 2 2 3 4 0 3 5 1 1 3	1.0000000 1.5000000 0.0000000 0.0000000 2.0000000
6 2 1 3 5 1 5 3 3 1 5 3 5 1 5 1 3 5 3 1	0.0000000 6.0000000 0.0000000 6.0000000 6.0000000 6.0000000

Problem E. Broken Glass

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

Little Serezha enjoys breaking glass. When he finds a glass, he starts throwing rocks at it one by one. Consider the glass to be a rectangle of size $W \times H$ such that its bottom left corner is located at the origin and top right corner has coordinates (W, H).

Serezha throws N stones one after another. When he throws stone i, it hits a point with coordinates (x_i, y_i) and four cracks appear on the glass going in four different directions from that point: left, up, right and down. Each crack goes either till the nearest crack perpendicular to it or till the edge of the glass. Interestingly, when Serezha hits an existing crack or an intersection of cracks, new cracks go only in the directions there is no crack yet. For example, when Serezha hits a horizontal crack (but no vertical crack), two new vertical cracks appear.

Unfortunately, when Serezha's mother finds out that he has broken another glass, she makes him repair it. And as Serezha is good only at breaking things, he asked you to help him count the number of pieces he has broken the glass into. A piece is a part of the initial rectangular glass bounded by cracks and/or edges of the glass and containing no cracks inside.

Input

The first line of input contains three integers W, H and N ($1 \le W, H \le 10^9$, $1 \le N \le 10^5$). Each of the following N lines contains a pair of coordinates. Line number $(i+1)$ contains two integers x_i and y_i: coordinates of the point at which i-th stone hit the glass ($0 \le x_i \le W$, $0 \le y_i \le H$).

Output

On the first line of output, print one integer: the number of pieces into which the glass is broken.

Examples

standard input	standard output
6 4 2 4 1 5 2	7
7 7 5 3 4 7 4 7 6 5 6 4 5	10

Problem F. Higgs Bozon

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

In this problem, we consider the total energy of the system of N nodes. Each of the nodes can exist in two different states. We will refer to the states as 0 and 1.

The total energy of the system is the sum of two values. The first one is made up of the unary potentials of each node: for each node and each of its states, we know the contribution of this node to the total energy in this state. Formally, we are given $U_i(v)$, the contribution of node v when it is in state i, for every possible i and v.

The second summand is the sum of binary potentials: for some pairs of nodes, an interaction between them takes place, so the contribution of this interaction to the total energy is known. Formally, we are given $U_{ij}(u, v)$, the contribution of the interaction between u and v when node u is in state i and node v is in state j. There are precisely M pairs of nodes for which interaction takes place.

There is also an important condition in the system: for each u and v,

$$U_{00}(u, v) + U_{11}(u, v) \leq U_{01}(u, v) + U_{10}(u, v).$$

Given the parameters described above, find the state of the system in which its total energy is minimal possible.

Input

The first line of input contains integers N and M: the number of nodes and the number of interactions ($1 \leq N \leq 1000$, $0 \leq M \leq 1000$).

Each of the following N lines contains the unary potentials of the nodes: for each v from 0 to $N - 1$, integers $U_0(v)$ and $U_1(v)$ are given on a separate line ($-10\,000 \leq U_i(v) \leq 10\,000$).

In the next M lines, interactions are described. Each of these lines contains six integers: u, v, $U_{00}(u, v)$, $U_{01}(u, v)$, $U_{10}(u, v)$ and $U_{11}(u, v)$ ($0 \leq u, v < n$, $-10\,000 \leq U_{ij}(u, v) \leq 10\,000$). For each pair of nodes, there is at most one interaction.

Output

On the first line of output, print one integer: the minimal total energy of the system. The second line must contain a sequence of N integers each of which is either 0 or 1. These integers must represent the states of nodes 0, 1, ..., $N - 1$ which allow the system to achieve minimal total energy.

Example

standard input	standard output
2 1 6 9 -2 4 0 1 10 5 2 -3	9 1 0

Problem G. DNF

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Do you know what a pleasure it is to minimize a disjunctive normal form on a warm summer evening? If you don't, you should definitely try it. In order to simplify your task, we will remind you what a disjunctive normal form is.

Consider a vector of boolean variables $(x_1, x_2, \ldots, x_n)$ and boolean functions of the form $f(x_1, x_2, \ldots, x_n)$.

A *simple conjunction* is a conjunction of one or more terms. Terms are either variables or their negations. An example of a simple conjunction is $x_1\overline{x}_2\overline{x}_4x_5$. This conjunction evaluates to 1 on all vectors in which there are ones at first and fifth positions and zeros at second and fourth positions; on all other vectors, it evaluates to 0.

A *disjunctive normal form* of a boolean function $f(x_1, x_2, \ldots, x_n)$ is $D(x_1, x_2, \ldots, x_n) = K_1 \vee K_2 \vee \ldots \vee K_r$ where all K_i are simple conjunctions and $D(x) = f(x)$ for every possible boolean vector $(\alpha_1, \alpha_2, \ldots, \alpha_n)$. The disjunction evaluates to 1 on all vectors on which at least one of the simple conjunctions K_1, K_2, ..., K_r evaluates to 1; on all other vectors, it evaluates to 0.

The *cardinality* of a simple conjunction is the number of variables used in it. A disjunctive normal form of f is *minimal* if the sum of cardinalities of all the conjunctions included in this form is the minimal possible.

Now it's time to check whether you understand the definitions above. You are supposed to find the minimal disjunctive normal form of a boolean function of N variables which equals 1 on all possible boolean vectors except two: one that consists only of ones $(1, 1, \ldots, 1)$ and the other one containing only zeros $(0, 0, \ldots, 0)$.

Input

The first line of input contains an integer n ($2 \le n \le 64$), the number of boolean variables in the vector $(x_1, x_2, \ldots, x_n)$.

Output

Output file should contain as many lines as there are simple conjunctions in the form you found, one line for each simple conjunction. Each conjunction must be printed in the form "p_1 `&` p_2 `&` ... `&` p_k" where each p_i is either of the form "j" in case the simple conjunction contains x_j or "`!`j" if it contains $\overline{x}_j$. Note that there is a single space before and after each '`&`' sign; no other spaces are possible. Simple conjunctions can be given in any order. Terms in each simple conjunction can also be given in any order.

Example

standard input	standard output
2	1 & !2 2 & !1

Problem H. Nothing to Do

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

When people have nothing to do, they sometimes do strange things. For example, when Alexey is sitting on a lecture in MSU, he always has nothing to do (it's obvious that he isn't going to listen to the professor, isn't it?). He decided to write some numbers on a piece of paper.

First, he wrote an infinite sequence of all non-negative integers in increasing order (the lecture was very long, you see).

Then he replaced this sequence by a new one in which each number is changed to the bitwise-xor-sum of this number and all numbers to the left of it in the old sequence. You know, sometimes it seems like there is no end to such torture as sitting in the class... He repeated this replace operation over and over again until suddenly, the lecture was over.

Now, Alexey is trying to remember some interesting numbers he got during that process. Your task is to help him find that numbers. For each number, he will tell you its index and the number of replaces he already did for that moment.

The *bitwise-xor-sum* of two integers a and b is an integer c such that i-th digit of c in its binary representation is 1 if a and b differ in this digit or 0 if they are equal.

Input

First line contains an integer n: the number of values Alexey asks you to find ($1 \le n \le 250\,000$).

Next n lines contain description of Alexey's numbers. Each description contains two integers r_i and p_i which mean he asks you to find p_i-th number in the sequence after he made r_i replaces ($0 \le r_i, p_i \le 10^{18}$). Elements of the sequence are numbered from 0.

Output

Output n lines. On i-th line, print the i-th number Alexey asked you to find.

Example

standard input	standard output
3 0 3 1 2 2 4	3 3 6

Note

Explanation for the sample test:

Before all replacement operations, the sequence is 0, 1, 2, 3, 4, 5, 6, 7, 8, ..., so the first interesting number is 3.

After the first replacement, the sequence is 0, 1, 3, 0, 4, 1, 7, 0, 8, ..., so the second interesting number is $0 \oplus 1 \oplus 2 = 3$.

After the second replacement, the sequence is 0, 1, 2, 2, 6, 7, 0, 0, 8, ..., so the third interesting number is $0 \oplus 1 \oplus 3 \oplus 0 \oplus 4 = 6$.

Problem I. Points

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

There are n *black* points and q *red* points on the plane. For each red point, find the distance from it to the nearest black point. In this problem, we define the *distance* between two points (x_1, y_1) and (x_2, y_2) as $\max\{|x_1 - x_2|, |y_1 - y_2|\}$.

The first red point is be given in the input. The position of each next point can be generated using the coordinates of previous point and the answer for it by the following formulae:

$$\begin{aligned} qx_{i+1} &= (|A \cdot qx_i + B \cdot answer_i + C| \bmod (2M+1)) - M \\ qy_{i+1} &= (|D \cdot qy_i + E \cdot answer_i + F| \bmod (2M+1)) - M \end{aligned}$$

Here, qx_i and qy_i are coordinates of i-th red point and $answer_i$ is the answer for this point (i. e. the distance from it to the nearest black point). Notation $|expression|$ means the absolute value of $expression$.

Input

The first line contains two integers n and q ($1 \le n \le 10^5$, $1 \le q \le 5 \cdot 10^5$).

The second line contains the integer M.

The third line contains three integers A, B and C.

The fourth line contains three integers D, E and F.

Next n lines contain coordinates of black points, one point per line.

The last line contains two integers qx_1 and qy_1 which are the coordinates of the first red point.

All coefficients (A, B, C, D, E, F, M) and coordinates of points do not exceed 10^8 by absolute value. M is guaranteed to be strictly positive.

Output

On the first line of output, print one integer: the sum of all $answer_i$.

Example

standard input	standard output
3 3 10 1 1 -4 -1 2 0 1 7 2 -4 -6 -9 0 1	19

Note

Explanation for the example above:

The first red point is $(0, 1)$. The answer is 5.

The second red point is $(-9, -1)$. The answer is 8.

The third red point is $(-5, 7)$. The answer is 6.

Problem J. Lines

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

We have a number of straight lines on the plane. We also have a rectangle with sides parallel to coordinate axes. We want to know how many pairs of lines intersect in the interior of the rectangle.

Input

The first line of input contains one integer n, the number of lines on the plane ($1 \le n \le 10^5$).

Each of the next n lines of input contains a description of a line on the plane as four integers x_1, y_1, x_2 and y_2 which are coordinates of two different points on that line.

Each of the last four lines of input contain two integers: coordinates of rectangle corners. Rectangle corners are given in clockwise order. Sides of the rectangle are parallel to coordinate axes.

It is guaranteed that all the given lines on the plane are different. All coordinates are integers and don't exceed 10^6 by absolute value. It is also guaranteed that all points of intersection are located on the distance of at least 10^{-6} from the border of the rectangle.

Output

On the first line of output, print one integer: the number of pairs of lines which intersect inside the given rectangle.

Examples

standard input	standard output
2 1 0 0 1 0 0 1 1 0 0 0 1 1 1 1 0	1
4 0 0 1 1 0 10 10 10 0 1 2 1 0 2 2 2 0 0 0 5 5 5 5 0	2

Problem K. k-megainversions

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Let us define a k-megainversion. For a given permutation P of numbers $\{1, 2, \ldots n\}$, a sequence of integers $1 \le i_1 < i_2 < \ldots < i_k \le n$ is called a k-megainversion if $P_{i_1} > P_{i_2} > \ldots > P_{i_k}$.

A permutation is called even if the number of 2-megainversions in it is even. Your task is to calculate the expected number of k-megainversions in an even permutation P. Suppose that P is chosen equiprobably.

Input

The only line of input contains two integers n and k ($1 \le n, k \le 30\,000$).

Output

On the first line of output, print one real number: the expected number of k-megainversions of an even permutation of size n.

Your answer will be considered correct if the absolute or relative error does not exceed 10^{-6}.

Examples

standard input	standard output
3 2	1.333333333333
3 3	0.000000000000