

The 6th Guangxi Collegiate Programming Contest

Contest Session

June 4, 2023

Problem List

A	Alpha, Beta, Omega
B	Brilliant Idea
C	Chasing Game
D	Dream Team
E	Evil Capitalist
F	Forever Young
G	GCD Spanning Tree
H	Hide and Seek
I	Indivisible Partition
J	June
K	Keyboard
L	Love Choice
M	Meteor Shower

**Note that for the submission of C++ language,
please choose the compiler version C++(g++7.5.0) .**

This problem set should contain 13 problems on 16 numbered pages.
Please inform a runner immediately if something is missing from your problem set.

Problem A. Alpha, Beta, Omega

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

Slay the Spire is a game about survival, collecting cards, and building a powerful deck to defeat your enemies. Recently, Colin has become interested in a special card called Alpha.

- Playing an Alpha card shuffles a Beta card into the draw pile.
- Playing a Beta card shuffles an Omega card into the draw pile.
- Playing an Omega card grants the final Omega buff, allowing Colin to defeat his enemies and win.



Unfortunately, Colin is always unlucky. He wants to know when victory is guaranteed, even under the worst possible sequence of shuffles.

The process is simplified as follows. Initially, the draw pile contains only n_a Alpha cards, n_b Beta cards, and n_o Omega cards. In each round:

- The entire draw pile is shuffled, and Colin draws the top k cards.
- If at least one of the drawn cards is an Omega card, Colin wins immediately.
- Otherwise, Colin plays all k drawn cards. Each played Alpha card is removed and adds one Beta card to the draw pile; each played Beta card is removed and adds one Omega card to the draw pile. Newly added cards cannot be drawn until a later round.

Find the minimum number of rounds after which Colin is guaranteed to have won, regardless of how unlucky the shuffles are.

Input

A single line contains four integers n_a , n_b , n_o , and k ($0 \leq n_a, n_b, n_o \leq 3 \times 10^8$, $1 \leq k \leq n_a + n_b + n_o$).

The first three integers are the initial numbers of Alpha, Beta, and Omega cards, respectively. Colin draws exactly k cards in each round.

Output

Output a single integer: the minimum number of rounds after which Colin is guaranteed to have won.

Examples

standard input	standard output
3 0 0 2	4
100000000 100000000 100000000 1	300000001

Note

In the first example, the worst case proceeds as follows:

First round: Colin may draw two Alpha cards and shuffle two Beta cards into the draw pile.

Second round: Colin may draw one Alpha card and one Beta card, then shuffle one Beta card and one Omega card into the draw pile.

Third round: Colin may draw two Beta cards and shuffle two Omega cards into the draw pile.

Fourth round: The draw pile now contains only three Omega cards, so Colin draws an Omega card and wins.

Therefore, in the worst case, Colin first draws an Omega card in the fourth round.

Problem B. Brilliant Idea

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

One day, a brilliant idea came to Colin's mind.

Consider a string S of length n , indexed from 0 to $n - 1$, that consists of lowercase English letters. A successful turn consists of the following operations:

- Choose two integers l and r such that $0 \leq l < r < n$ and the substring $S_l S_{l+1} \dots S_r$ contains at least two distinct characters.
- Choose an index p with $l \leq p \leq r$, and replace every character S_i for $l \leq i \leq r$ with the character S_p .

For example, if $S = \text{abc}$ and Colin chooses $l = 1$, $r = 2$, and $p = 2$, then S becomes acc .

The process ends when no successful turn is possible. Let ans_S be the maximum number of successful turns that can be performed starting from S . If Colin can perform successful turns indefinitely, then ans_S is infinite.

Among all strings of length n consisting of lowercase English letters, count:

1. the strings for which $ans_S = 0$;
2. the strings for which $ans_S > 0$ and ans_S is finite;
3. the strings for which ans_S is infinite.

Input

The only line contains a single integer n ($2 \leq n \leq 10^9$), the length of the strings.

Output

Output three integers: the numbers of strings in the three categories above, in the same order. Output each number modulo $10^9 + 7$.

Example

standard input	standard output
2	26 650 0

Note

There are 26 strings of length 2 with $S_0 = S_1$, and each of them has $ans_S = 0$.

Each of the remaining $26 \cdot 25 = 650$ strings has two distinct characters. Exactly one successful turn can be performed, so each of them has $ans_S = 1$.

Problem C. Chasing Game

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

Colin and Eva are playing a chasing game on a tree with n nodes numbered from 1 to n .

In each game, Eva starts at node S at time 0 and moves toward her destination T , traversing one edge per second along the unique simple path from S to T . More precisely, if this path is $p_0, p_1, \dots, p_k$, where $p_0 = S$ and $p_k = T$, then Eva is at node p_i at time i . After reaching T , she remains there until Colin catches her.

Colin starts at node S' at time 0. During each second, he may either remain at his current node or traverse one edge, arriving at the chosen node at the end of that second. Colin catches Eva when they occupy the same node at the same integer time.

The two players repeat the game independently for q different choices of S , T , and S' . For each game, determine the earliest time at which Colin can catch Eva and the node where the catch occurs.

Input

The first line contains two integers n and q ($2 \leq n \leq 2 \times 10^5$, $1 \leq q \leq 2 \times 10^5$) — the number of nodes and the number of games.

Each of the next $n - 1$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), denoting an edge between nodes u_i and v_i . The given edges form a tree.

Each of the next q lines contains three integers S_i , T_i , and S'_i ($1 \leq S_i, T_i, S'_i \leq n$, $S_i \neq T_i$), denoting Eva's starting node, Eva's destination, and Colin's starting node, respectively, for the i -th game.

Output

For each game, output two integers t_i and p_i : the earliest time t_i at which Colin can catch Eva and the node p_i where the catch occurs.

It can be proved that the value of p_i is unique for every game.

Example

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

Problem D. Dream Team

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

Summer training is coming soon, and in order to achieve good results in the upcoming contests, everyone wants to find their favorite teammates to form their own dream team!

There are $3n$ students in the training team with corresponding numbers 1 to $3n$, and they will form n teams, each consisting of 3 students.

It is now known that there are m pairs of students (x_i, y_i) who are **directly familiar** with each other (i.e. x_i is directly familiar with y_i , and y_i is directly familiar with x_i).

Friends of friends are friends. We define that student a is **familiar** with student b if one of the following conditions is met:

1. a is **directly familiar** with b , or
2. There exists a series of students $x_1, x_2, \dots, x_k$ ($k > 0$) satisfies:
 - a is **directly familiar** with x_1
 - for all $1 \leq i < k$, x_i is **directly familiar** with x_{i+1}
 - x_k is **directly familiar** with b

As a retired contestant, Colin clearly knows that it is important for teammates to be familiar with each other. We define a student's anxiety level as the number of students in his team that he is **not familiar with** (of course everyone is familiar with himself). Now Colin wants you to find a plan satisfies: divide all $3n$ students into n teams, and each student is in exactly one team, and each team has exactly 3 students, and minimize the sum of anxiety level of all $3n$ students.

You only need to tell Colin the minimum sum of anxiety level of all $3n$ students.

Input

The first line contains two integers n, m ($1 \leq n \leq 10^6, 1 \leq m \leq 2 \times 10^5$), representing there are $3n$ students and m pairs of students **directly familiar** with each other.

For the following m lines, each line contains two integers x_i, y_i ($1 \leq x_i, y_i \leq 3 \times n, x_i \neq y_i$), representing x_i, y_i are **directly familiar** with each other.

Output

Output one integer: the minimum possible sum of the anxiety levels of all $3n$ students.

Example

standard input	standard output
3 5 1 2 1 3 4 5 4 6 7 8	4

Problem E. Evil Capitalist

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

Today, Colin and Eva learned about exploitation in their “Basic Principles of Marxism” course. That night, Colin dreamed that he was a poor worker being exploited by the capitalist Eva without even realizing it.

Suppose you are an evil capitalist who employs n exploited workers, numbered from 1 to n . Worker i currently earns a_i yuan. Under your strict supervision, the workers cannot tell one another their salaries. You thought this arrangement would last forever, but the workers have begun to feel exploited and are planning to strike!

Without their labor, you cannot maintain your luxurious lifestyle. You therefore decide to raise some salaries to stop the protest. Initially, every worker is **unsatisfied**. You must arrange two-person chats until either **all workers are satisfied** or **exactly one worker remains unsatisfied**. In the latter case, nobody else remains to join that worker in a protest.

To arrange a chat, choose two **unsatisfied** workers x and y and temporarily allow them to reveal their salaries to each other:

- If $a_x = a_y$, both workers feel that they have been treated equally and become **satisfied**.
- If $a_x > a_y$, worker x becomes **satisfied**. Worker y , however, realizes that they have been treated differently. You must increase worker y ’s salary to a_x , paying $a_x - a_y$ yuan and setting $a_y = a_x$. Worker y remains **unsatisfied**.
- Similarly, if $a_y > a_x$, worker y becomes **satisfied**. You must pay worker x an additional $a_y - a_x$ yuan and set $a_x = a_y$, while worker x remains **unsatisfied**.

As an evil capitalist, you want to spend as little as possible. Find the minimum total amount of money required to achieve your goal.

Input

The first line contains one integer n ($1 \leq n \leq 2 \times 10^5$), the number of workers.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$), where a_i is the current salary of worker i .

Output

Output a single integer: the minimum amount of money you need to spend.

Example

standard input	standard output
4 100 101 1 101	1

Note

First, let workers 1 and 2 talk. Worker 2 becomes satisfied, and you pay worker 1 one yuan, making $a_1 = 101$.

Then let workers 1 and 4 talk. Because their salaries are equal, they both become satisfied.

Only worker 3 remains unsatisfied, so the goal has been achieved.

Problem F. Forever Young

Input file: standard input
Output file: standard output
Time limit: 7 seconds
Memory limit: 512 megabytes

I remember what someone said to me a long time ago, "The path you choose, even if you walk on your knees, you have to finish it." Friends, although the world is getting more and more impetuous, as long as we can persist in our efforts for the sake of our once pure dreams and moving motives, we can remain true to ourselves all the way, no matter what others do.

As a retired contestant, Colin hopes that all participants will always have a young heart and an eternal love for algorithms and competitive programming.

Colin has a tree T with n vertices, numbered from 1 to n . Vertex 1 is the root.

Vertex u is an ancestor of vertex v if and only if u lies on the path from v to the root. In particular, every vertex is its own ancestor.

Let subtree $_u$ be the set of all vertices v such that u is an ancestor of v .

Colin and Eva assign two integer weights, c_u and e_u , to every vertex u . Define

$$ans_u = \left(\sum_{x \in \text{subtree}_u} \sum_{y \in \text{subtree}_u} \min\{|c_x - c_y|, |e_x - e_y|\} \right) \bmod (10^9 + 7).$$

The two sums range over ordered pairs (x, y) ; consequently, pairs with $x \neq y$ are counted in both orders. Calculate ans_u for every vertex u .

Input

The first line contains a single integer n ($1 \leq n \leq 5 \times 10^5$), the number of vertices.

Each of the next $n - 1$ lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$), denoting an edge between vertices u_i and v_i . The given edges are guaranteed to form a tree.

Each of the next n lines contains two integers c_i and e_i ($1 \leq c_i, e_i \leq 10^9$), the two weights of vertex i .

Output

Output n lines.

The i -th line must contain the integer ans_i .

Example

standard input	standard output
5	44
1 2	12
1 3	0
2 4	0
2 5	0
9 5	
2 8	
7 1	
4 3	
6 6	

Problem G. GCD Spanning Tree

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

You are given a connected undirected graph G with n vertices and m edges. Edge i has a positive integer weight w_i .

In one operation, you may choose one edge and change its weight to any positive integer.

A spanning tree T of G is called an **x -spanning tree** if the greatest common divisor (gcd) of the weights of all edges in T is divisible by x .

For a given x , Colin wants to know the minimum number of operations required so that G contains at least one **x -spanning tree**.

There are k independent queries. Every query refers to the original graph: the operations considered for one query are not actually performed and do not affect any other query.

Recall that a **spanning tree** of G is a connected, acyclic subgraph that contains all n vertices; it therefore has exactly $n - 1$ edges.

Input

The first line contains three integers n , m , and k ($2 \leq n \leq 10^5$, $1 \leq m, k \leq 10^5$), the number of vertices, edges, and queries, respectively.

Each of the next m lines contains three integers u_i , v_i , and w_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq w_i \leq 10^6$), denoting an edge of weight w_i between vertices u_i and v_i . The graph is guaranteed to be connected.

Each of the next k lines contains one integer x_i ($1 \leq x_i \leq 10^6$), describing the i -th query.

Output

Output k lines.

The i -th line must contain the minimum number of operations required for the i -th query.

Example

standard input	standard output
4 6 3	0
1 2 2	1
1 3 5	2
1 4 4	
2 3 9	
2 4 12	
3 4 6	
2	
3	
5	

Problem H. Hide and Seek

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Eva is playing a special game of “hide and seek” with Colin. Unlike the traditional game, Eva chooses a hidden positive integer x for Colin to guess.

First, Eva tells Colin a positive integer a and guarantees that $x \leq a$.

Then Eva gives Colin m constraints. Constraint j consists of two non-negative integers b_j and c_j and requires

$$b_j \bmod x = c_j.$$

Colin wants to enumerate every possible value of x , but he does not know how many there are. Determine the number of positive integers $x \leq a$ that satisfy all the constraints.

It is guaranteed that at least one such integer exists.

Input

The first line contains two integers m and a ($1 \leq m \leq 2 \times 10^5$, $1 \leq a \leq 10^9$), the number of constraints and the upper bound on x , respectively.

Each of the next m lines contains two integers b_j and c_j ($0 \leq c_j \leq b_j \leq 10^9$), describing the constraint $b_j \bmod x = c_j$.

Output

Output one integer: the number of positive integers that satisfy all the given constraints.

Example

standard input	standard output
2 10 7 1 8 2	2

Problem I. Indivisible Partition

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 32 megabytes

Please note the unusual memory limit.

Eva calls a string S of length n (indexed from 0 to $n - 1$) **divisible** if both of the following conditions hold:

- $n > 1$, and
- there exists an integer $k \in [1, n)$ such that $k \mid n$ and $S_i = S_{i \bmod k}$ for every integer $i \in [0, n)$.

Here S_i denotes the character of S at index i . For example, if $S = \text{abcde}$, then $S_0 = \text{a}$ and $S_1 = \text{b}$.

A string is **indivisible** if it is not divisible. For example, `abcde` and `a` are indivisible, whereas `aa` is divisible.

Colin defines an **indivisible partition** of a string S of length n as a sequence of integers $\{p_1, p_2, \dots, p_k\}$ ($k \geq 2$) satisfying:

- $p_1 = 0$ and $p_k = n$, and
- for every integer $i \in [2, k]$, $p_{i-1} < p_i$ and the substring $S[p_{i-1} : p_i - 1]$ is indivisible.

Here $S[l : r]$ denotes the substring of S from index l through index r , inclusive. For example, if $S = \text{abcde}$, then $S[1 : 3] = \text{bcd}$.

Given a string S , count its distinct indivisible partitions modulo 998244353.

Two partitions $\{p_1, p_2, \dots, p_{k_1}\}$ and $\{p'_1, p'_2, \dots, p'_{k_2}\}$ are considered different if $k_1 \neq k_2$, or if there exists an integer $i \in [1, k_1]$ such that $p_i \neq p'_i$.

Input

The input contains a string S ($1 \leq |S| \leq 5 \times 10^3$) consisting of lowercase English letters.

Output

Output a single integer: the number of distinct indivisible partitions of S , modulo 998244353.

Example

standard input	standard output
aab	3

Note

The three distinct indivisible partitions of `aab` are:

1. $\{0, 1, 2, 3\}$, which divides the string into `a + a + b`;
2. $\{0, 1, 3\}$, which divides the string into `a + ab`;
3. $\{0, 3\}$, because `aab` itself is indivisible.

Problem J. June

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

June is the sixth month of the Gregorian calendar and has 30 days. It marks the beginning of astronomical summer in the Northern Hemisphere and astronomical winter in the Southern Hemisphere.

June is named after Juno, the Roman goddess of youth and protection. Her Latin name, *Iūnō*, comes from a root meaning “young” and is associated with vitality and fertility.

June is a month of youth and vitality. Many wonderful events take place in June, including the national college entrance examination, Colin’s birthday, and, of course, the 6th Guangxi Collegiate Programming Contest!

A person named S will participate in an event named T this June.

Print the sentence “Good luck to S in T and have fun!” to wish them good luck.

Input

The first line contains a string S ($1 \leq |S| \leq 50$) consisting of lowercase English letters: the person’s name.

The second line contains a string T ($1 \leq |T| \leq 50$) consisting of lowercase English letters: the event’s name.

Output

Output one line: “Good luck to S in T and have fun!” (without quotation marks).

Example

standard input	standard output
you gxcpc	Good luck to you in gxcpc and have fun!

Problem K. Keyboard

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

Colin enjoys chatting with Eva online, but typing on a traditional keyboard feels too tiring. He therefore buys a new keyboard, shown below, with only the two keys he really needs: Copy and Paste.



Colin's new keyboard.

Suppose that the string S is currently displayed on the screen. The keys behave as follows (note that this differs from a regular keyboard):

- Pressing **Copy** replaces the string stored in memory, denoted by T , with the entire current string S .
- Pressing **Paste** appends T to the current string, so the displayed string becomes $S + T$. Before the first Copy operation, T is the empty string.

Even this keyboard requires too much thought, so Colin decides to press the two keys alternately. If he performs any operations, the first one must be Copy, the second Paste, the third Copy, and so on. He may stop after any operation.

Initially, the screen displays S_1 and T is empty. Can Colin obtain the target string S_2 by following the rule above?

Input

The first line contains a string S_1 ($1 \leq |S_1| \leq 2 \times 10^5$) consisting of lowercase English letters: the string initially displayed on the screen.

The second line contains a string S_2 ($1 \leq |S_2| \leq 2 \times 10^5$) consisting of lowercase English letters: Colin's target string.

Output

If Colin can obtain the target string, output "Smart People's Big Win!" (without quotation marks).

Otherwise, output "Lazy Dog's Great Failure..." (without quotation marks).

Examples

standard input	standard output
colin colinloveseva	Lazy Dog's Great Failure...
heihei heiheiheihei	Smart People's Big Win!

Problem L. Love Choice

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

It's the beginning of a new relationship. Will this one result in joy or sorrow? In LoveChoice, you choose the actions that determine how the relationship unfolds. Pick your responses to each of the situations. These endearing and bittersweet stories are surprisingly realistic and easy to relate to.



To win Eva's heart, Colin studies hard by playing LoveChoice on Steam. To simplify the problem, we describe it as follows:

There are n multiple-choice questions; the i -th question has a_i normal choices. Additionally, some questions have an alternative hidden choice. There is only one correct choice for each question, and if a hidden choice exists, it must be the correct choice.

In a round of the game, you must answer all the questions in order. **After answering all the questions**, the system will tell you if **each question** is answered correctly.

Colin is calm. For each question, he will only choose one of the choices that may not be the wrong choice, with equal probability each time. However, Colin's observation ability is relatively ordinary, so if a question i has a hidden choice, Colin will not be aware of its existence before knowing that all the a_i normal choices for this question are wrong.

Colin is hardworking. His goal is to answer all questions correctly in a round. Please calculate the expected number of rounds for him to **answer all the questions correctly** for the first time.

Input

The first line has two integers n, m ($1 \leq m \leq n \leq 5 \times 10^3$), representing that there are n questions in the game and that there are m questions that have a hidden choice.

The second line has n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$), representing that there are a_i normal choices for the i -th question.

The third line has m distinct integers $b_1, b_2, \dots, b_m$ ($1 \leq b_i \leq n$), representing that the b_i -th question has a hidden choice.

Output

Output a single integer, representing the expected number of rounds for him to **answer all the questions correctly** for the first time, modulo $10^9 + 7$.

Example

standard input	standard output
2 1 1 3 1	333333338

Note

In the example, there are 2 questions with 1 choice and 3 choices, respectively, and question 1 has a hidden choice. Thus, Colin cannot answer the first question correctly in the first round, as he does not know the existence of the hidden choice in the first round.

The probability of answering all the questions correctly for the first time in the second round is $\frac{2}{3}$.

The probability of answering all the questions correctly for the first time in the third round is $\frac{1}{3}$.

So the answer to the example is $\frac{2}{3} \times 2 + \frac{1}{3} \times 3 = \frac{7}{3}$, which modulo $10^9 + 7$ is 333333338.

Problem M. Meteor Shower

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

Colin hears that there will be a meteor shower in a few days. Being a romantic boyfriend, he plans to take Eva to watch it.

We model the situation on a two-dimensional plane. The Earth is a circle centered at the origin O with radius r ; its surface consists of the points satisfying $x^2 + y^2 = r^2$.

The meteor shower contains n meteors. Meteor i appears at the point (x_i, y_i) and then moves continuously along the ray starting at this point in the direction of the nonzero vector (dx_i, dy_i) . It keeps moving in this direction unless it collides with the Earth. Meteors do not collide with one another, so their movements are independent.

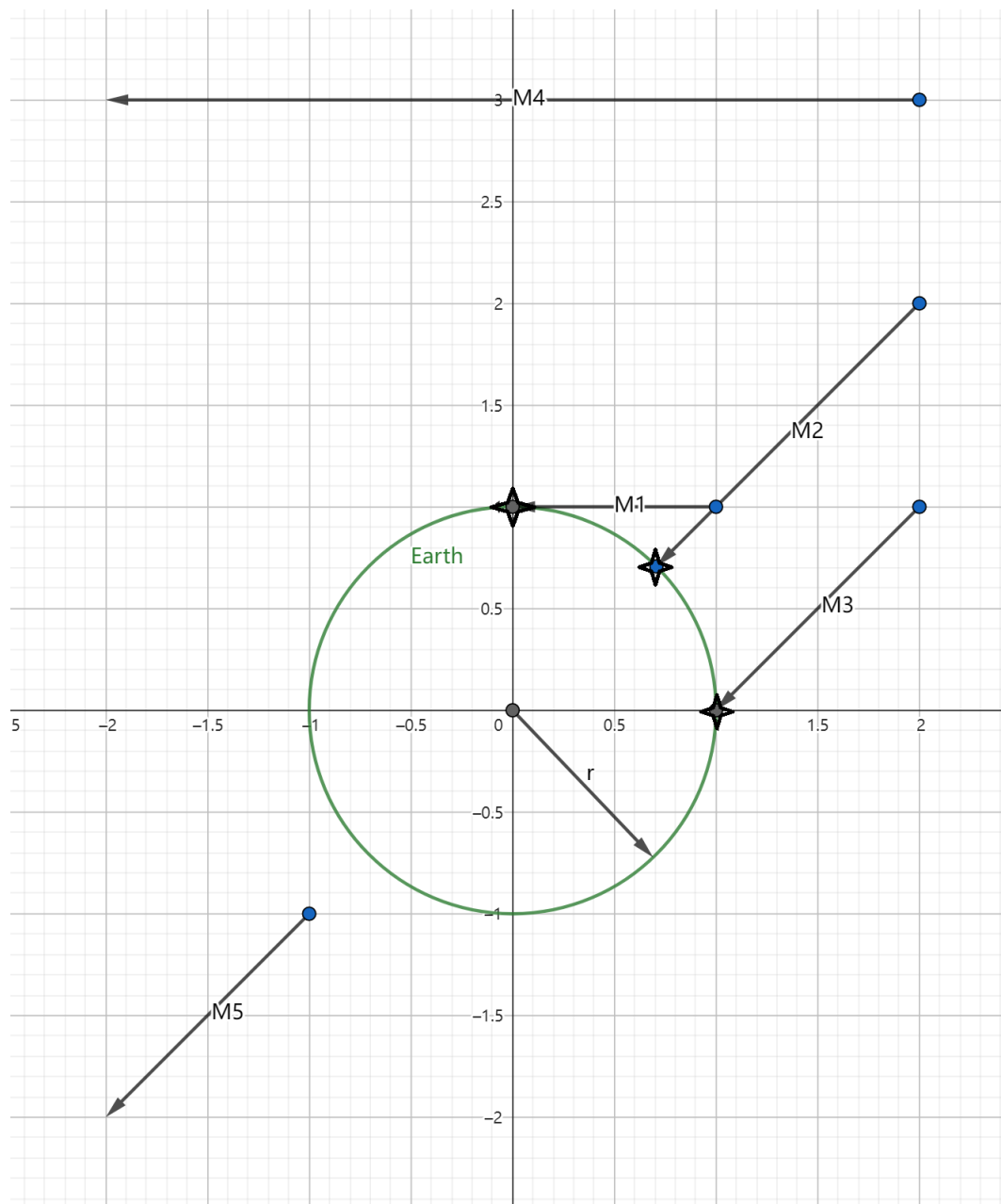


Illustration of the example.

For a meteor that never collides with the Earth, the optimal observation site is the point on the Earth's surface with minimum distance to the meteor's trajectory ray. If a meteor does collide with the Earth—touching the surface counts as a collision—Colin instead wants the first point at which it reaches the surface.

For every meteor, determine which case occurs:

- If it collides with the Earth, output “Crash at (x, y) ”, where (x, y) is its first collision point.
- Otherwise, output “Observe at (x, y) ”, where (x, y) is the optimal observation site.

Print every coordinate with exactly four digits after the decimal point.

Input

The first line contains two integers n and r ($1 \leq n \leq 10^5$, $1 \leq r \leq 10^9$): the number of meteors and the radius of the Earth.

Each of the next n lines contains four integers x_i , y_i , dx_i , and dy_i ($-10^9 \leq x_i, y_i, dx_i, dy_i \leq 10^9$). The pair (x_i, y_i) is meteor i 's starting point, and (dx_i, dy_i) is its direction vector. It is guaranteed that $(dx_i, dy_i) \neq (0, 0)$.

It is also guaranteed that the distance from O to every meteor's starting point is at least $r + 0.1$.

Output

Output n lines. Line i must have one of the following forms:

- “Crash at (x, y) ” if meteor i collides with the Earth, where (x, y) is its first collision point;
- “Observe at (x, y) ” otherwise, where (x, y) is the optimal observation site.

Do not print the quotation marks. Print both coordinates with exactly four digits after the decimal point. A coordinate is accepted if its absolute error from the jury answer does not exceed 10^{-3} .

Example

standard input	standard output
5 1	Crash at (0.0000,1.0000)
1 1 -1 0	Crash at (0.7071,0.7071)
2 2 -1 -1	Crash at (1.0000,0.0000)
2 1 -1 -1	Observe at (0.0000,1.0000)
2 3 -1 0	Observe at (-0.7071,-0.7071)
-1 -1 -1 -1	

Note

The answers for the example are illustrated in the figure in the statement.