The 3rd Universal Cup

Stage 22: Zhengzhou

December 21-22, 2024

This problem set should contain 13 problems on 20 numbered pages.

Based on

China Collegiate Programming Contest (CCPC)

Problem A. A + B = C Problem

Time limit: 1 second
Memory limit: 1024 megabytes

Given three positive integers p_A, p_B, p_C, Bobo challenges you to find out three infinite binary strings A, B, C with period p_A, p_B and p_C respectively satisfying $A \oplus B = C$, or determine it is impossible to do so.

Please refer to the Note section for the formal definition of period and exclusive or.

Input

The first line of the input contains a single integer T ($1 \le T \le 10^4$), denoting the number of test cases. The description of the test cases follows.

The first and the only line of each test case contains three integers p_A, p_B and p_C ($1 \le p_A, p_B, p_C \le 10^6$).

It is guaranteed that the sum of $\max(p_A, p_B, p_C)$ over all test cases does not exceed 10^6.

Output

For each test case, output "NO" (without quotes) in one line if no solution exists. Otherwise, output "YES" (without quotes) in one line. Then, output three binary strings of length p_A, p_B and p_C in three lines, denoting the first p_A, p_B, p_C character(s) of the infinite strings A, B, C respectively.

You can output "YES" and "NO" in any case (for example, strings "yES", "yes", and "Yes" will all be recognized as a positive response).

Example

standard input	standard output
2 2 3 6 2 3 5	YES 01 011 001110 NO

Note

Let $s = s_1 s_2 s_3 \ldots$ and $t = t_1 t_2 t_3 \ldots$ be infinite binary strings.

The period of s is the smallest positive integer k satisfying $s_i = s_{i+k}$ for all $i \ge 1$.

The exclusive or of strings s and t is given by $s \oplus t$ satisfying $(s \oplus t)_i = s_i \oplus t_i$ for all $i \ge 1$.

Problem B. Rolling Stones

Time limit: 1 second
Memory limit: 1024 megabytes

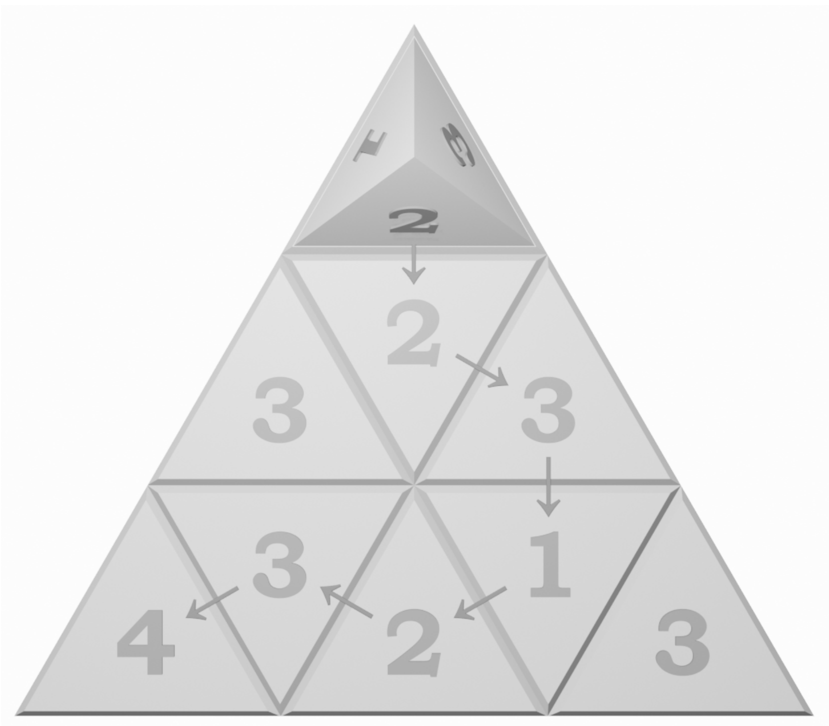
Bobo has been playing a puzzle called *Rolling Stones*, which takes place on an equilateral triangular board consisting of n ($n \geq 2$) rows and n^2 cells. Each cell on the board is labeled with a number from 1 to 4. Bobo also has a tetrahedral stone, with each face numbered from 1 to 4 (a tetrahedral dice), initially placed at the first cell in the first row of the board. The position of the stone is as follows: the face with the number 1 is towards the left, the face with the number 2 is towards the next row, the face with the number 3 is towards the right, and the face with the number 4 is on the bottom side.

The goal of the puzzle is to roll the stone to a target cell under the following rules:

- **Matching Numbers**: When the stone rests on a cell, the number on the cell must match the number on the stone's bottom face.
- **Single Visit**: Each cell can only be visited once throughout the journey, including the starting and target cells.

The stone rolls by tipping along an edge that touches the board, moving it to a neighboring cell. Given the board layout, the target cell, and the stone's initial orientation, Bobo wants to know: is it possible to reach the target cell following the rules? If possible, what is the minimum number of rolls required to reach the target?

The illustration for a solution of the first sample test is given as follows.



Illustration

Input

The first line contains an integer n ($2 \leq n \leq 100$), denoting the size of the board.

Then n lines follows, with the i-th ($1 \leq i \leq n$) line containing $2i - 1$ numbers $a_{i,1}, a_{i,2}, \ldots, a_{i,2i-1}$, where each $1 \leq a_{i,j} \leq 4$ indicates the number on the j-th cell from left to right in the i-th row. It is guaranteed that $a_{1,1} = 4$.

Then another line follows, containing two integers x, y ($2 \le x \le n, 1 \le y \le 2x-1$). Here, (x, y) represents the target cell, located at the y-th cell from left to right in the x-th row.

Output

If there is no way to roll the stone to the target cell, output -1 in a line. Otherwise, output the minimum number of rolls to roll the stone to the target cell in a line.

Examples

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

Problem C. Middle Point

Time limit: 1 second
Memory limit: 1024 megabytes

Bobo is exploring a set of lattice points on a two-dimensional plane. Initially, the set of points is defined as $S = \{(0,0),(A,0),(0,B),(A,B)\}$. Bobo's goal is to include a specific lattice point (X,Y) in S. To achieve the goal, Bobo may perform the following operation:

- Select two lattice points $P, Q \in S$ such that $\frac{P+Q}{2}$ is also a lattice point, and add $\frac{P+Q}{2}$ to S.

Your task is to help Bobo find a sequence of operations that minimizes the number of steps to achieve the goal or determine if it is impossible to do so.

Input

The first line of the input contains two integers A and B ($0 \le A, B \le 10^9$), describing the parameters of the initial lattice points.

The second line of the input contains two integers X and Y ($0 \le X \le A$, $0 \le Y \le B$), denoting the coordinates of the target lattice point.

Output

If it is impossible to achieve the goal, output -1 in one line. Otherwise, output a single integer k ($0 \le k \le 10^5$) in one line, denoting the total number of operations to perform. Then k lines follow. The i-th line contains four integers U_i, V_i, S_i, T_i ($0 \le U_i, V_i, S_i, T_i \le 10^9$), describing the lattice points $P = (U_i, V_i)$ and $Q = (S_i, T_i)$ chosen in the i-th operation. If there exist multiple solutions, output any.

Examples

standard input	standard output
2 2 1 1	1 0 0 2 2
8 8 5 0	3 0 0 8 0 4 0 8 0 4 0 6 0
0 0 0 0	0
2024 0 1012 0	1 0 0 2024 0
2024 2024 2023 2023	-1
8 6 7 3	3 0 0 8 0 4 0 8 0 6 0 8 6

Problem D. Guessing Game

Time limit: 2 seconds
Memory limit: 1024 megabytes

Alice and Bobo are engaged in an intriguing game, and you have the honor of being the judge.

In the k-th round, you will write down a pair of integers (a_k, b_k) on the blackboard, clearly visible to both Alice and Bobo. Once they see the numbers, you will secretly select an integer $1 \le i \le k$, giving a_i to Alice and b_i to Bobo. The excitement builds as Alice and Bobo take turns either claiming they know the other's number or admitting that they do not know the answer, starting with Alice. The player who correctly guesses the other's number first will win the game!

Both players are exceptionally smart and honest, making the game all the more captivating. As you observe their interactions, you can't help but wonder: in the k-th round, how many values of i that Alice wins, and how many values of i that Bobo wins?

Input

The first line contains a single integer q ($1 \le q \le 10^6$), denoting the total number of integer pairs.

Each of the following q lines contains a single pair of integers (a_k, b_k) ($1 \le a_k, b_k \le 10^5$).

It is guaranteed that $(a_1, b_1), (a_2, b_2), \ldots, (a_q, b_q)$ are distinct.

Output

Output q lines. In the k-th line, output two integers A_k and B_k, denoting the number of i that Alice wins and Bobo wins respectively.

Example

standard input	standard output
4 1 1 1 2 2 1 2 2	1 0 0 2 1 2 0 0

Problem E. Permutation Routing

Time limit:	1 second
Memory limit:	1024 megabytes

Bobo is given a tree $T = (V, E)$ of n vertices, where there is a number p_i on vertex i initially, and $p_1, p_2, \ldots, p_n$ is a permutation of 1 to n, meaning that all integers from 1 to n appear exactly once in $p_1, p_2, \ldots, p_n$

In each operation, Bobo can select a matching $M \subseteq E$ (M is a matching means that no two edges in M share a common vertex), and for each $(u, v) \in M$, swap the number on vertex u and vertex v (i.e. swap p_u and p_v).

Bobo wants to use at most $3n$ operations to make $p_i = i$ for each $1 \le i \le n$, can you please help him?

Input

There are multiple test cases. The first line of the input contains an integer T ($T \ge 1$), indicating the number of test cases. For each test case:

The first line contains a single integer n ($1 \le n \le 1000$) — the number of vertices of the tree.

The second line contains n integers $p_1, p_2, \cdots, p_n$ ($1 \le p_i \le n$, and p is a permutation of 1 to n) — the initial number on vertex i is p_i.

Then follow $n - 1$ lines, each with integers u, v ($1 \le u, v \le n, u \ne v$) — meaning that there is an edge between u and v.

It is guaranteed that the sum of n^2 of all test cases will not exceed 10^6.

Output

For each test case: The first line contains a single integer m ($0 \le m \le 3n$) — the number of operations you used.

Then m lines follow, where each line starts with an integer $0 \le k_i < n$ — denoting the number of edges in the matching you select in the i-th operation. Then k_i integers $t_{i,1}, t_{i,2}, \cdots, t_{i,k_i}$ follow, denoting the indexes of edges you select.

Example

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

Problem F. Infinite Loop

Time limit: 1 second
Memory limit: 1024 megabytes

Bobo is trapped in an infinite time loop of a peculiar day! Each day consists of exactly k hours, and every day, n tasks arrive for Bobo to complete.

- The i-th task of the day arrives at the beginning of the a_i-th hour and requires b_i hours of uninterrupted effort to finish.
- Bobo works diligently and always follows a disciplined approach: whenever there are unfinished tasks, **Bobo works on the earliest received unfinished task**.

At the beginning of the first day, Bobo starts with no tasks.

Your mission is to help Bobo answer q queries. For the i-th query, you are given x_i, the day on which a task is received, and y_i, the index of the task received on that day. Your goal is to determine the exact day and hour when Bobo will complete the y_i-th taskof day x_i.

Input

The first line contains three space-separated integers, which are n ($1 \le n \le 10^5$), k ($1 \le k \le 10^8$), and q ($1 \le q \le 10^5$), respectively.

The next n lines each contain two space-separated integers, where the i-th line contains a_i ($1 \le a_i \le k$) and b_i ($1 \le b_i \le k$). It is guaranteed that a_i is strictly monotonically increasing.

Then q lines follow, each containing two space-separated integers, where the i-th line contains x_i ($1 \le x_i \le 5 \times 10^5$) and y_i ($1 \le y_i \le n$).

Output

Output q lines, where the i-th line outputs two space-separated integers d_i and h_i, indicating that the task for the i-th query is completed at the h_i-th hour on the d_i-th day.

Examples

standard input	standard output
2 5 6 1 1 4 3 1 1 1 2 2 1 2 2 3 1 3 2	1 1 2 1 2 2 3 1 3 2 4 1
3 10 5 2 4 3 1 10 7 2 2 7 1 4 3 5 2 28 3	3 1 8 10 6 2 6 7 34 10

Problem G. Same Sum

Time limit: 2 seconds
Memory limit: 1024 megabytes

Bobo is working with an integer sequence $a_1, a_2, \ldots, a_n$ of length n. He must process q queries in order. Each query is of one of the following two types:

- $1\ L\ R\ v$ ($1 \le L \le R \le n$, $0 \le v \le 2 \cdot 10^5$): for all $i \in [L, R]$, update $a_i \leftarrow a_i + v$;
- $2\ L\ R$ ($1 \le L < R \le n$, $R - L + 1$ is even): determine if elements $a_L, a_{L+1}, \ldots, a_R$ can be divided into $(R - L + 1)/2$ pairs of integers with the same sum.

Your task is to help Bobo process these queries efficiently.

Input

The first line of input contains two integers n, q ($1 \le n, q \le 2 \cdot 10^5$).

The second line of input contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 2 \cdot 10^5$).

Then q lines follow. Each of the following lines contains a query, described in the statement.

Output

For each query of the second type, output "YES" (without quotes) in one line if elements $a_L, a_{L+1}, \ldots, a_R$ can be divided into $(R - L + 1)/2$ pairs of integers with the same sum; otherwise, output "NO" (without quotes) in one line.

You can output "YES" and "NO" in any case (for example, strings "yES", "yes", and "Yes" will all be recognized as a positive response).

Example

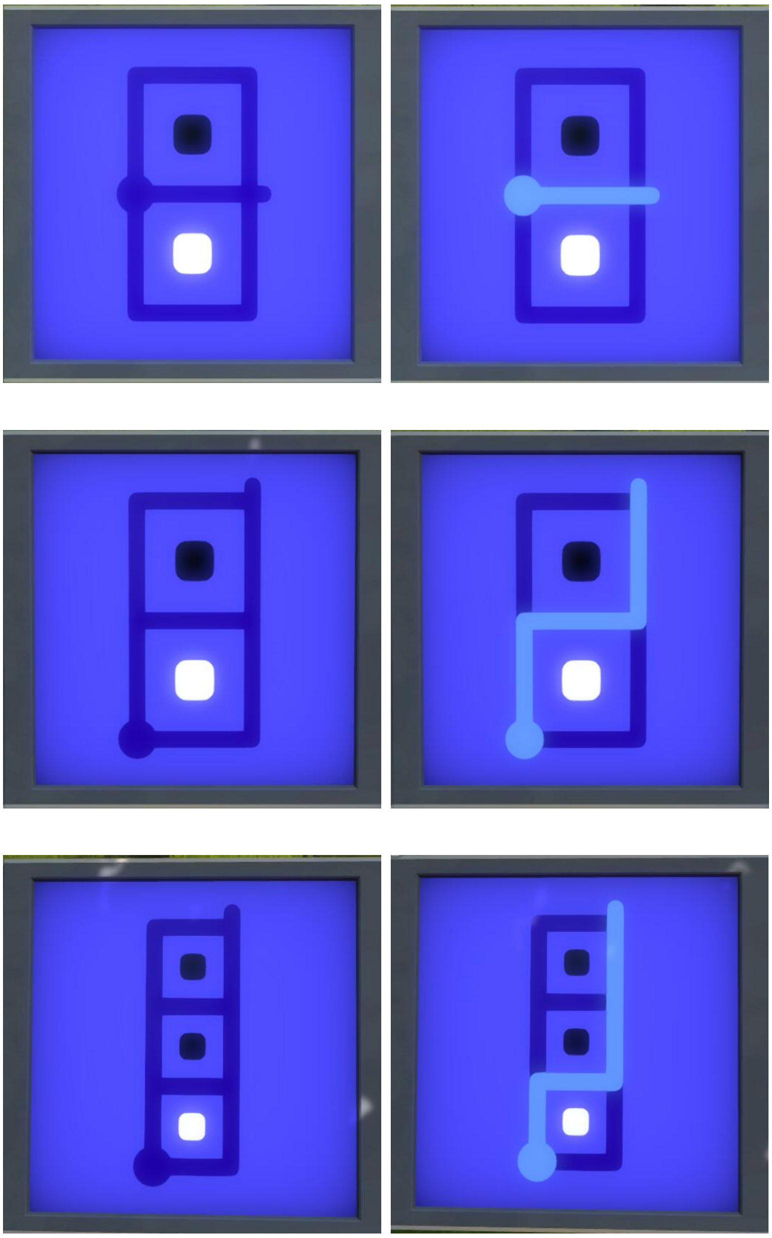
standard input	standard output
8 4 1 2 3 4 5 6 7 8 2 1 8 1 1 4 4 2 1 6 2 1 8	YES NO YES

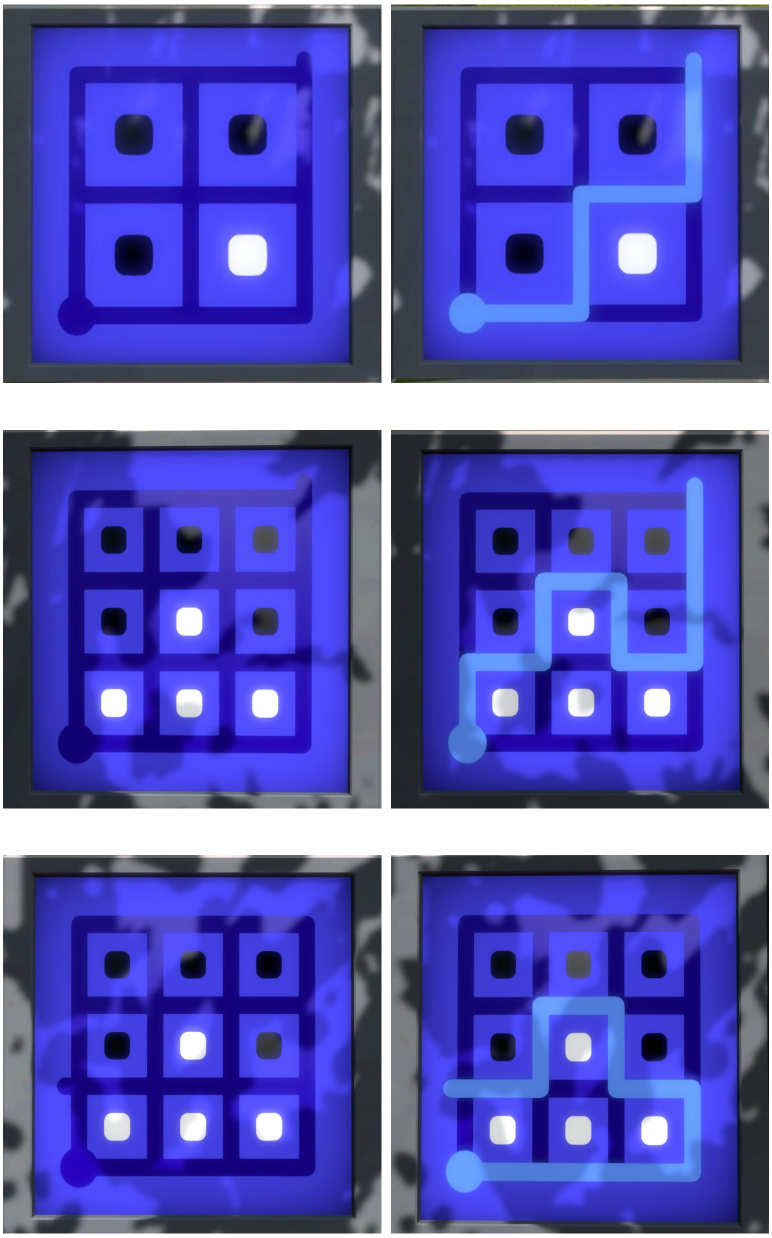
Problem H. The Witness

Time limit: 1 second
Memory limit: 1024 megabytes

During his spare time, Bobo likes to play puzzle games a lot. His favorite one is *The Witness*, an acclaimed 2016 puzzle video game designed by Jonathan Blow. The Witness contains 9 principal types of puzzles and several hidden "environmental" puzzles, totaling an amount of 664 puzzles spread over the open world of the game. A player is invited to explore the world and to deduce the rules of various puzzles they encounter.

Among all types of puzzles, Bobo expertizes at a certain type called the "Black and White Squares" puzzle. This kind of puzzle is based on a rectangular grid and requires the player to connect a starting point to an end point with a simple grid path while splitting the grid cells of two different colors. The following are some real examples of this type of puzzle in-game together with one of their solutions. The starting point and the end point are marked with circles and a protruding semicircle from the grid, respectively.





Given an instance of the "Black and White Squares" puzzle **such that all cells are either black or white**, Bobo decides to leave you the challenge: can you provide a solution to the puzzle or assert there are none?

Formally, an instance of the "Black and White Squares" puzzle is described by the following:

- two positive integers n, m, representing the number of rows and columns of the rectangular grid. There are then $n \times m$ cells and $(n+1) \times (m+1)$ vertices in the grid. We label the vertex on the upper-left corner of the grid as $(0, 0)$, the vertex on the lower-right corner of the grid as (n, m), and the rest accordingly.
- An $n \times m$ two-dimensional array, representing the color of each **cell**. **The color of each cell can only be either black or white.**
- A starting point (sx, sy) and an end point (ex, ey) , each belonging to one of the **vertices** on the **border** of the grid. Here a vertex (x, y) is on the **border** means at least one of the following is satisfied:
  - $x = 0$

  - $x = n$
  - $y = 0$
  - $y = m$

  Also, **the starting point and the end point cannot coincide.**

A solution to the "Black and White Squares" puzzle is described by a path $P = ((x_1, y_1), (x_2, y_2), \ldots, (x_\ell, y_\ell))$ ($\ell \geq 2$) satisfying the following properties:

- $(x_1, y_1) = (sx, sy)$ and $(x_\ell, y_\ell) = (ex, ey)$.
- For each $2 \leq i \leq \ell$, it follows (x_{i-1}, y_{i-1}) and (x_i, y_i) are **adjacent** on the grid, i.e. one of the following is satisfied:
  - $x_i = x_{i-1}$ and $|y_i - y_{i-1}| = 1$
  - $|x_i - x_{i-1}| = 1$ and $y_i = y_{i-1}$
- P is simple, i.e., for each $1 \leq i < j \leq \ell$, either $x_i \neq x_j$ or $y_i \neq y_j$.
- **Each region of the grid separated by the path contains only one color.**

Input

The first line contains two integers n, m ($1 \leq n, m \leq 40$).

Then n lines follow, the i-th line contains a string s_i of length m consisting of only 'B' and 'W', where the j-th character of s_i represents the color of the cell on the i-th row and j-th column of the grid.

Then a line consisting of four integers sx, sy, ex, ey ($0 \leq sx, ex \leq n, 0 \leq sy, ey \leq m$) follows, denoting the starting point and the end point. **It is guaranteed that both the starting point and the end point are on the border of the grid and they don't coincide.**

Output

If there is no solution to the given "Black and White Squares" puzzle instance, output "`NO`" (without quotes) in a line.

Otherwise, output "`YES`" (without quotes) in the first line. Then output an integer ℓ ($\ell \geq 2$), denoting the number of vertices contained in the solution path. Then on the ith of the next ℓ lines, output two integers (x_i, y_i), denoting the i-th vertex in the solution path. Your answer will be considered correct if it meets the required conditions. If there are multiple solutions, you are allowed to output any of them.

You can output "`YES`" and "`NO`" in any case (for example, strings "`yES`", "`yes`", and "`Yes`" will all be recognized as a positive response).

Examples

standard input	standard output
3 3 BBB BWB WWW 3 0 0 3	YES 9 3 0 2 0 2 1 1 1 1 2 2 2 2 3 1 3 0 3
1 1 W 0 0 1 1	YES 3 0 0 1 0 1 1
2 2 WB BW 0 0 2 2	NO

Note

The first sample test describes the "Black and White Squares" puzzle instance in the fifth group of pictures in the statement.

For the second sample test, another valid solution path is $P = ((0,0),(0,1),(1,1))$.

Problem I. Best Friend, Worst Enemy

Time limit: 1 second
Memory limit: 32 megabytes

Bobo is analyzing a group of n people, where the i-th ($1 \le i \le n$) person has two attributes, x_i and y_i. The attribute values of different people are distinct. For any two persons $1 \le i, j \le n$ ($i \ne j$), Bobo defines their *friend index* $\text{Friend}(i, j)$ and *enemy index* $\text{Enemy}(i, j)$ respectively as follows:

$$\text{Friend}(i,j) \triangleq \max(|x_i - x_j|, |y_i - y_j|), \quad \text{Enemy}(i,j) \triangleq |x_i - x_j| + |y_i - y_j|.$$

For any $1 \le i, j \le n$ ($i \ne j$), Bobo calls the j-th person is a *best friend* of the i-th person if

$$\text{for all } 1 \le k \le n \ (k \ne i), \quad \text{Friend}(i,k) \ge \text{Friend}(i,j).$$

Also, for any $1 \le i, j \le n$ ($i \ne j$), Bobo calls the j-th person is a *worst enemy* of the i-th person if

$$\text{for all } 1 \le k \le n \ (k \ne i), \quad \text{Enemy}(i,k) \le \text{Enemy}(i,j).$$

Now Bobo wants to find out, for each $1 \le t \le n$, how many ordered pairs (i, j) satisfy $1 \le i, j \le t$, $i \ne j$, and the j-th person is both a best friend and a worst enemy of the i-th person if we only consider the first t people.

Please be aware of the unusual memory limit.

Input

The first line contains one integer, n ($2 \le n \le 4 \times 10^5$).

The next n lines each contain two space-separated integers, where the i-th line contains x_i and y_i ($1 \le x_i, y_i \le 10^7$). It is guaranteed that for $i \ne j$, either $x_i \ne x_j$ or $y_i \ne y_j$.

Output

Output n lines, each line containing a single integer, denoting the number of pairs that meet the requirements.

Examples

standard input	standard output
2 1 5 1 10	0 2
4 2 5 5 3 5 7 8 5	0 2 4 4
9 3 4 3 6 4 3 4 7 5 5 6 3 6 7 7 4 7 6	0 2 1 0 4 5 6 7 8
13 3 5 4 4 4 5 4 6 5 3 5 4 5 5 5 6 5 7 6 4 6 5 6 6 7 5	0 2 4 7 2 2 5 2 2 3 3 4 4

Note

In the first example, when considering only the first person, there are no ordered pairs that meet the requirements. When considering the first two people, there are two ordered pairs that meet the requirements: $(1,2)$ and $(2,1)$.

In the second example, when considering only the first person, there are no ordered pairs that meet the requirements. When considering the first two people, there are two ordered pairs that meet the requirements: $(1,2)$ and $(2,1)$. When considering the first three people, there are four ordered pairs that meet the requirements: $(1,2)$, $(1,3)$, $(2,1)$, and $(3,1)$. When considering the first four people, there are four ordered pairs that meet the requirements: $(2,1)$, $(2,4)$, $(3,1)$, and $(3,4)$.

Problem J. Balance in All Things

Time limit: 6 seconds
Memory limit: 1024 megabytes

Bobo is participating in a weird tournament with $2n$ players, labeled 1 through $2n$. Initially, all players have a score of zero. The tournament consists of k rounds, and in each round, players are paired for one-on-one matches.

The scoring mechanism is as follows: after each match, the player with the higher score loses 1 point, while the player with the lower score gains 1 point. If two players have the same score, the player with the lower label (i.e., the smaller number) is considered the winner and gains 1 point, while the other player loses 1 point.

To ensure balance and to make the tournament more exciting, the host decided that the absolute value of any player's score must never exceed 3 at any point in the tournament. Given these rules, Bobo wants to determine the number of possible ways to arrange the matches over the k rounds.

As the answer might be too large, you should output the answer modulo P, which is a specified prime number.

Input

The first line of input contains three integers n, k, P ($1 \le n \le 400, 1 \le k \le 20, 10^8 \le P \le 10^9 + 9$), whose meaning is already clear in the statement.

It is guaranteed that P is a prime.

Output

Output an integer in one line, denoting the answer.

Examples

standard input	standard output
3 1 1000000007	15
100 3 1000000007	894710378
6 6 1000000007	103387851
2 6 998244353	729

Problem K. Brotato

Time limit: 1.5 seconds
Memory limit: 1024 megabytes

Bobo is playing a game called **Brotato**. The game consists of n levels, each of which he can either pass or fail. Each level has a probability p of failure and a probability $1 - p$ of passing. If Bobo fails a level, he must normally restart from the first level.

Bobo is quite frustrated about the fact that each time he dies, he has to start over from the very beginning. Therefore, Bobo decided to cheat. Now, Bobo has k special items that allow him to continue from the same level after a failure rather than restarting from the beginning.

Given this setup, determine the minimum expected number of attempts for levels needed for Bobo to complete all n levels.

Input

The first line contains two integers n, k ($1 \le n \le 10^5, 0 \le k \le 10^9$), denoting the number of levels and the number of items, respectively.

The second line contains a number p ($0 < p \le 0.5$). It is guaranteed that p has at most 4 decimal places.

It is guaranteed that $np \le 20$.

Output

Output a number in a line denoting the answer.

Your answer is considered correct if its absolute or relative error doesn't exceed 10^{-9}. Namely, if your answer is a, and the jury's answer is b, then your answer is accepted if $\frac{|b-a|}{\max(b,1)} \le 10^{-9}$.

Examples

standard input	standard output
5 0 0.5	62.0000000000
5 1 0.5	47.0000000000
10000 0 0.002	247489700298.2536834329
100000 10 0.0002	38767507133.2322179824

Problem L. Z-order Curve

Time limit: 1 second
Memory limit: 1024 megabytes

Welcome to the China Collegiate Programming Contest (CCPC) Zhengzhou onsite! Bobo has noticed that the initials of "Zheng" and "Zhou" are both Z. This motivates him to study the well-known Z-order curve.

To introduce the Z-order curve, we first introduce the Moser–de Bruijn sequence $(B_t)_{t \geq 0}$, the ordered sequence of numbers whose binary representation has nonzero digits only in the even positions. The first few terms of the Moser-de Bruijn sequence are $0, 1, 4, 5, 16, 17, 20, 21$.

Each non-negative integer z can be uniquely decomposed into the sum of B_x and $2B_y$. Therefore, we can write down all natural numbers in an infinitely large table. The Z-order curve is then obtained by connecting all the numbers in numerical order.

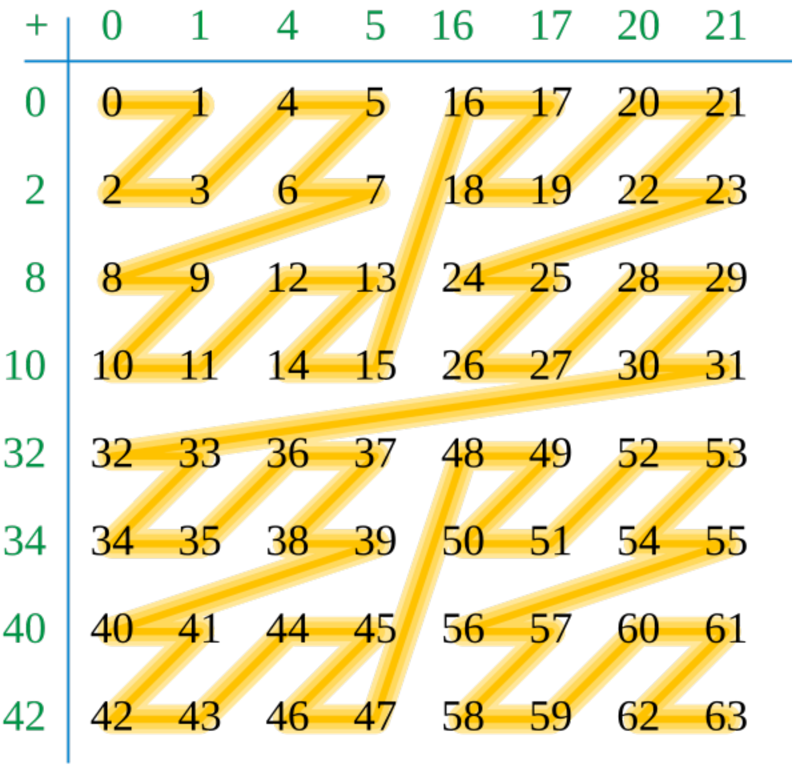


Illustration of Z-curve

Bobo now challenges you with the following problem: For a given fragment extracted from the Z-curve from L to R, find the smallest integer l such that the Z-curve from l to $l + R - L$ is identical to the given fragment (i.e., the curve from l to $l + R - L$ can be obtained by translating the curve from L to R).

Please note that in this problem, the curve is directed. Specifically, the curve from 1 to 2 is NOT identical to the curve from 3 to 4.

Input

The first line of the input contains a single integer T ($1 \leq T \leq 100$), denoting the number of test cases.

The first and only line of each test case contains two integers L and R ($0 \leq L < R \leq 10^{18}$).

Output

For each test case, output the answer in one line.

Example

standard input	standard output
4 17 20 0 63 38 40 998244353998244353 998244853998244853	1 0 6 2145186925057

Note

The following figure illustrates the Z-curve for the first and third test cases in the sample.

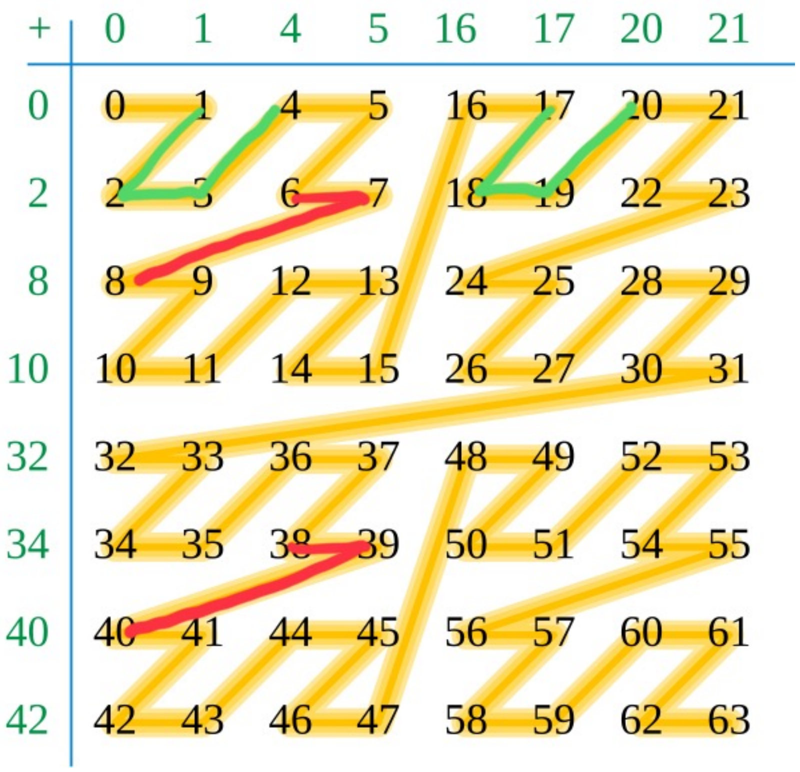


Illustration of test cases in the sample

(red: test case 1, green: test case 3)

Problem M. Rejection Sampling

Time limit: 1 second
Memory limit: 1024 megabytes

Bobo wants to use a rejection sampling algorithm to construct a random set $T \subset \{1, 2, \ldots, n\}$ of size k. For parameters $p_1, p_2, ..., p_n$ $(0 \le p_i \le 1)$ and integer k, the rejection sampler is defined as follows:

1. Initialize $T \leftarrow \emptyset$;
2. For each i $(1 \le i \le n)$, add i into T with probability p_i;
3. Output T if the size of T is exactly k; otherwise, repeat the process.

Now you are given integers $a_1, a_2, ..., a_n$ and k. Bobo needs to set the parameters $p_1, p_2, \ldots, p_n$ satisfying

- $\sum_{i=1}^{n} p_i = k$;
- for all $S \subseteq \{1, 2, \cdots, n\}$ such that $|S| = k$, the probability that the rejection sampler outputs S is proportional to $\prod_{i \in S} a_i$.

Your task is to find out the parameters $p_1, p_2, \ldots, p_n$ for Bobo. It is guaranteed that such parameters exist and **are unique**. Your answer will be considered correct if the absolute error of each p_i doesn't exceed 10^{-6} compared to the unique answer.

Input

The first line of the input contains two integers n and k $(2 \le n \le 10^5, 1 \le k \le n-1)$.

The second line of the input contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$.

Output

Output n lines. The i-th line contains a single real number p_i.

Your answer is considered correct if the absolute error of each parameter does not exceed 10^{-6}. Namely, if your answer is a, and the jury's answer is b, then your answer is accepted if $|b - a| \le 10^{-6}$ for all parameters.

Examples

standard input	standard output
3 2 5 5 5	0.666666666667 0.666666666667 0.666666666667
2 1 1 4	0.333333333333 0.666666666667
4 2 1 2 3 4	0.310035697652 0.473324044845 0.574114878920 0.642525378583