The 2025 icpc Wuhan Invitational Contest

Contest Session

April 27, 2025

Problem List

A	Problem Setting
B	Black Red Tree
C	One Must Imagine Sisyphus Happy
D	Odd and Even
E	Colorful Graph
F	Knapsack
G	Path Summing Problem
H	WildFire, This Is for You!
I	Bingo 3
J	Dictionary
K	Las Vegas
L	Subsequence
M	Flight Tracker

This problem set should contain 13 (thirteen) problems on 18 (eighteen) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.
https://sua.ac/

Hosted by

Problem Set Prepared by

It's against the rules to open non-contest websites during the contest.
If you're interested (which is our pleasure),
please scan the QR code only after the contest.

Problem A. Problem Setting

Members of the SUA Programming Contest Problem Setter Team are preparing problems for the 2025 ICPC Wuhan Invitational Contest. The problem they're currently working on has n properties to indicate its various aspects, such as difficulty, length of code, etc. The value of its i-th property is a_i.

The members have also proposed q suggestions, where the i-th suggestion can be denoted as three integers p_i, l_i, and r_i, which means the value of the p_i-th property should lie between l_i and r_i (both inclusive).

BaoBao is the author of the problem, and he is going to modify the problem according to these suggestions. He can spend one unit of time to increase or decrease the value of a property by 1. Calculate the smallest amount of time he needs so that all suggestions are satisfied, or report that it is impossible to do so.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 100$) indicating the number of test cases. For each test case:

The first line contains two integers n and q ($1 \le n, q \le 100$), indicating the number of properties and the number of suggestions.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 10^9$), where a_i is the value of the i-th property.

For the following q lines, the i-th line contains three integers p_i, l_i, and r_i ($1 \le p_i \le n$, $1 \le l_i \le r_i \le 10^9$), indicating that the value of the p_i-th property should lie between l_i and r_i (both inclusive).

Output

For each test case, output one line containing one integer, indicating the smallest amount of time needed to satisfy all suggestions. If it is impossible to do so, output `-1` instead.

Example

standard input	standard output

3
4 3
20 25 4 27
3 5 7
1 10 15
3 2 6
1 2
7
1 3 5
1 9 9
1 2
7
1 3 9
1 4 15
	6
	-1
	0

Note

For the first sample test case, BaoBao can change the 1-st property to 15 and the 3-rd property to 5. The answer is $(20 - 15) + (5 - 4) = 6$.

For the third sample test case, as $3 \le 7 \le 9$ and $4 \le 7 \le 15$, all suggestions are already satisfied and BaoBao does not need to change any property.

Problem B. Black Red Tree

Please note the UNUSUAL MEMORY LIMIT of this problem.

BaoBao has a tree with n vertices. Initially, all edges of the tree are black. BaoBao likes painting and counting in the tree, so he will perform $(n-1)$ painting operations. For the i-th operation, he will paint the q_i-th edge red. It's obvious that after all operations, all edges will become red.

Your task is to calculate how many simple paths in the tree contain exactly k black edges after each operation. Note that we consider the simple path from vertex u to v and from vertex v to u to be the same path.

Recall that a simple path is a path that does not go through the same edge multiple times.

Input

There is only one test case in each test file.

The first line of the input contains two integers n and k ($2 \le n \le 2 \times 10^5$, $1 \le k \le 10$) indicating the number of vertices in the tree and the number of black edges we count for paths.

For the following $(n-1)$ lines, the i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) indicating that the i-th edge in the tree connects vertices u_i and v_i.

For the following $(n-1)$ lines, the i-th line contains an integer q'_i ($1 \le q'_i < n$), indicating the **encoded** index of the i-th edge BaoBao paints. The real value of q_i is equal to $((q'_i + a_{i-1}) \bmod (n-1)) + 1$, where a_{i-1} is the answer to the $(i-1)$-th query. Specifically, we define a_0 as the number of simple paths in the tree that contain exactly k black edges before BaoBao paints. With the encoded queries, you're forced to calculate the answer to each query before processing the next one. It's guaranteed that $q_1, q_2, \ldots, q_{n-1}$ is a permutation of $1, 2, \ldots, n-1$.

Output

Output $(n-1)$ lines where the i-th line contains an integer indicating the number of simple paths with exactly k black edges after the i-th operation.

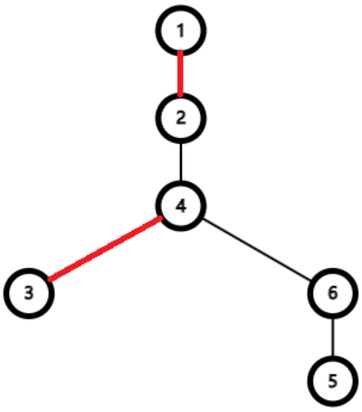
Examples

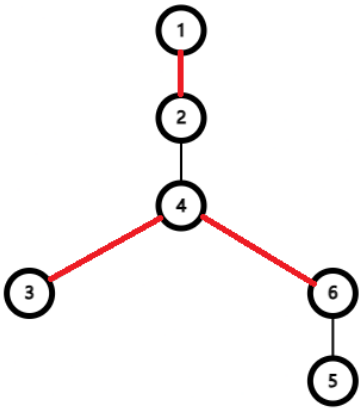
standard input	standard output

6 1
4 6
4 2
4 3
5 6
1 2
4
2
1
2
3
	5
	7
	8
	8
	0

6 2
4 6
4 2
4 3
5 6
1 2
2
5
5
4
3
	5
	4
	2
	0
	0

Note





For the first sample test case, the real values of q_1, q_2, q_3, q_4, q_5 are $5, 3, 4, 1, 2$, respectively. The tree after the first two operations is shown on the left. Black edges are represented with thinner lines, while red edges are represented with thicker lines.

Let $u \to v$ be the simple path from vertex u to v. The simple paths containing exactly one black edge are: $1 \to 3$, $1 \to 4$, $2 \to 3$, $2 \to 4$, $3 \to 6$, $4 \to 6$, $5 \to 6$.

For the second sample test case, the real values of q_1, q_2, q_3, q_4, q_5 are $3, 1, 5, 2, 4$, respectively. The tree after the first three operations is shown on the right.

The simple paths containing exactly two black edges are: $1 \to 5$, $2 \to 5$.

Problem C. One Must Imagine Sisyphus Happy

Sisyphus has been appointed to cultivate a long strip of farmland.

The farmland is divided into n cells from left to right, numbered from 1 to n. At the beginning, all cells are overgrown with weeds, and Sisyphus stands at cell 1. He will walk from cell 1 to cell n, and then walk back to cell 1. It takes him 1 hour to check and remove the weeds he finds in each cell he passes (regardless of whether there are actually weeds in that cell). The time taken to move between two cells is negligible, so one round of checking takes him $(2n - 1)$ hours, during which each of the first $(n - 1)$ cells will be checked twice and the n-th cell will only be checked once. After each round of checking, he needs to report how many cells of weeds have been removed, and the time taken for the report is also negligible.

However, under the farmland lies an endless supply of weed seeds. After Sisyphus removes the weeds in cell i and leaves that cell, the weeds will regrow after a_i hours and 1 minute. Specifically, if Sisyphus happens to be checking that cell again when the weeds regrow, the newly grown weeds will also be removed.

Camus wants to know, for each $1 \le k \le m$, how many cells of weeds Sisyphus needs to remove during his k-th round of checking. If Sisyphus already knows the numbers, he should be happy.

Input

There are multiple test cases. The first line of the input contains an integer T $(1 \le T \le 2 \times 10^3)$ indicating the number of test cases. For each test case:

The first line contains two integers n and m $(1 \le n, m \le 10^6)$, indicating the number of cells and the number of rounds.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(0 \le a_i \le 10^9)$. The weeds in the i-th cell will regrow after a_i hours and 1 minute.

It's guaranteed that neither the sum of n nor the sum of m of all test cases will exceed 10^6.

Output

For each test case output one line containing m integers separated by a space, where the i-th integer indicates how many cells of weeds will be removed during the i-th round of checking.

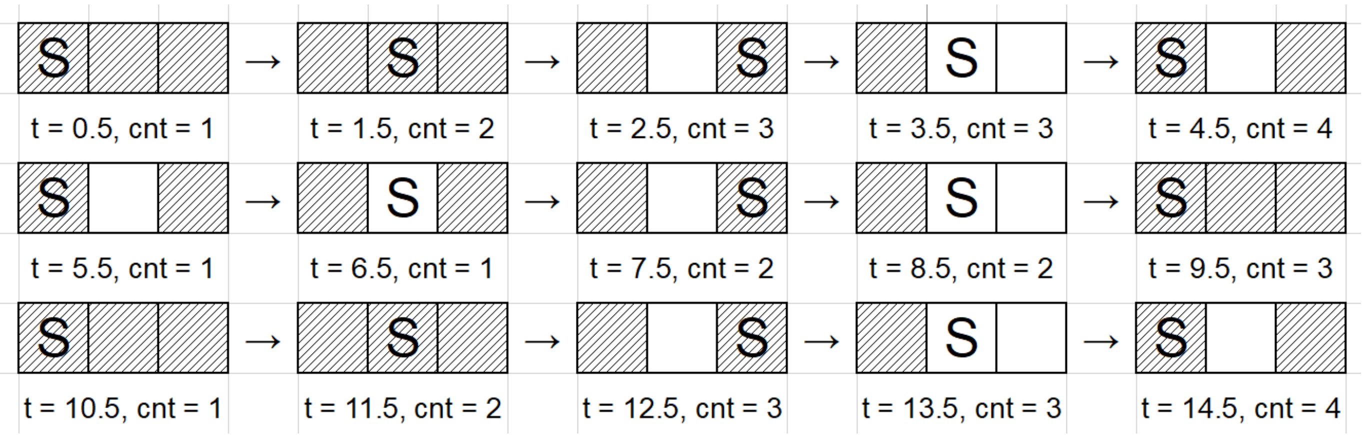
Example

standard input	standard output

3
3 3
0 7 1
5 4
6 16 10 5 10
3 3
2 1 20
	4 3 4
	6 3 4 3
	5 3 3

Note

We illustrate the first sample test case below. The empty cells indicate the cells without weeds, the shaded cells indicate the cells with weeds (including the weeds currently being removed), `S` indicates Sisyphus, `t` indicates the current time in hours, and `cnt` indicates how many cells of weeds have been removed in this round (including the weeds currently being removed).



Problem D. Odd and Even

Given an integer sequence $A = a_1, a_2, \cdots, a_k$ of length k, you can split it into several non-empty contiguous subarrays such that each element belongs to exactly one subarray. For each subarray, we calculate the sum of its elements. Let p be the number of subarrays with an odd sum, and q be the number of subarrays with an even sum.

You need to answer m queries about this sequence. Each query is denoted as an integer r, and you need to find the maximum possible values of p and q under the condition that $p + q = r$, respectively.

As sequence A might be long, we'll describe it in run-length encoding. More formally, given n pairs of integers $(v_1, l_1), (v_2, l_2), \cdots, (v_n, l_n)$, sequence A is derived as follows: starting from an empty sequence, first append v_1 to the back of the sequence l_1 times, then append v_2 to the back of the sequence l_2 times, ..., finally append v_n to the back of the sequence l_n times. See the explanation of the sample test case below for an example.

Input

There is only one test case in each test file.

The first line contains two integers n and m ($1 \le n, m \le 2 \times 10^5$), indicating the length of the run-length encoding of the sequence, and the number of queries.

For the following n lines, the i-th line contains two integers v_i and l_i ($1 \le v_i, l_i \le 10^9$). Therefore, the length of sequence A can be calculated as $k = \sum\limits_{i=1}^{n} l_i$. It is guaranteed that $v_i \ne v_{i+1}$ for all $1 \le i < n$.

For the following m lines, the i-th line contains an integer r_i ($1 \le r_i \le k$), indicating the i-th query.

Output

For each query, output one line containing two integers separated by a space, indicating the maximum possible values of p and q under the condition that $p + q = r$, respectively.

Example

standard input	standard output

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

Note

For the sample test case, $A = 5, 5, 5, 2, 2, 7$.

For the second query, we need to split the sequence into 2 contiguous subarrays.

- To maximize the number of subarrays with odd sums, we can split A into $5, 5, 5 \,|\, 2, 2, 7$. Both subarrays have odd sums, so the maximum possible p is 2.
- To maximize the number of subarrays with even sums, we can split A into $5, 5 \,|\, 5, 2, 2, 7$. Both subarrays have even sums, so the maximum possible q is 2.

For the fifth query, we need to split the sequence into 5 contiguous subarrays.

- To maximize the number of subarrays with odd sums, we can split A into $5 \,|\, 5 \,|\, 5 \,|\, 2, 2 \,|\, 7$, where 4 of them have odd sums, so the maximum possible p is 4.
- To maximize the number of subarrays with even sums, we can split A into $5, 5 \,|\, 5 \,|\, 2 \,|\, 2 \,|\, 7$, where 3 of them have even sums, so the maximum possible q is 3.

Problem E. Colorful Graph

Given an undirected graph with n vertices and m edges, you need to paint each edge with one of the m colors, such that for any vertex u and any color c, at most 2 edges connecting vertex u are painted with color c.

More formally, for all integers $1 \le u \le n$, the following constraint must be satisfied: let d_u be the number of edges connecting vertex u, let $e_{u,1}, e_{u,2}, \cdots, e_{u,d_u}$ be these edges, and let $w(e)$ be the color of edge e (colors are numbered from 1 to m). Then, for all integers $1 \le c \le m$, c appears at most twice in the sequence $w(e_{u,1}), w(e_{u,2}), \cdots, w(e_{u,d_u})$.

Let f_u be the number of different colors on all edges connecting vertex u. Find a way to paint the edges and minimize $\sum\limits_{u=1}^{n} f_u$.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($2 \le n \le 2 \times 10^5$, $1 \le m \le \min(\frac{n(n-1)}{2}, 2 \times 10^5)$), indicating the number of vertices and the number of edges in the graph.

For the following m lines, the i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$), indicating that the i-th edge connects vertices u_i and v_i.

It's guaranteed that there are no self loops or multiple edges in the graph. However, the given graph might not be connected. It's also guaranteed that neither the sum of n nor the sum of m of all test cases will exceed 2×10^5.

Output

For each test case, output one line containing m integers $w_1, w_2, \cdots, w_m$ ($1 \le w_i \le m$) separated by a space, where w_i is the color painted on the i-th edge.

If there are multiple valid answers, you can output any of them.

Example

standard input	standard output

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

Problem F. Knapsack

There are n groups of items, where the i-th group contains a_i items, and each item has a weight of 2^{b_i}.

There are also m knapsacks, each with a capacity of k. Calculate the smallest k such that it is possible to put all $\sum\limits_{i=1}^{n} a_i$ items into the knapsacks, and the total weight of the items in each knapsack does not exceed k.

Note that each item should be placed into exactly one knapsack. One knapsack may contain items from different groups, and the items from the same group may go into different knapsacks.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n \le 2 \times 10^5$, $1 \le m \le 10^9$) indicating the number of groups of items and the number of knapsacks.

For the following n lines, the i-th line contains two integers a_i and b_i ($1 \le a_i \le 10^9$, $0 \le b_i \le 10^9$), where a_i is the number of items in the i-th group, and 2^{b_i} is the weight of each item in the i-th group.

It's guaranteed that the sum of n of all test cases does not exceed 2×10^5.

Output

For each test case, output one line containing one integer indicating the answer. As the answer may be large, output it modulo 998 244 353. Note that the modulus here is only to simplify the output, and you need to minimize the answer before the modulus.

Example

standard input	standard output

2
5 4
3 0
2 3
3 1
1 3
2 1
2 20250427
1000000000 1000000000
114514 1919810
	10
	628956724

Problem G. Path Summing Problem

There is a grid with n rows and m columns. Each cell of the grid has an integer in it, where $a_{i,j}$ indicates the integer in the cell located at the i-th row and the j-th column.

Let (i, j) be the cell located at the i-th row and the j-th column. You now start from $(1, 1)$ and need to reach (n, m). When you are in cell (i, j), you can either move to its right cell $(i, j + 1)$ if $j < m$ or move to its bottom cell $(i + 1, j)$ if $i < n$.

Let $\mathbb{S}$ be the set consisting of integers in each cell on your path, including $a_{1,1}$ and $a_{n,m}$. The value of a path is defined as the number of elements in $\mathbb{S}$ (recall that sets do not contain duplicated elements). For all possible paths, calculate the sum of their values.

Input

There are multiple test cases. The first line of the input contains an integer T $(1 \le T \le 10^3)$ indicating the number of test cases. For each test case:

The first line contains two integers n and m $(1 \le n, m \le 10^5, 1 \le n \times m \le 10^5)$ indicating the number of rows and columns of the grid.

For the following n lines, the i-th line contains m integers $a_{i,1}, a_{i,2}, \cdots, a_{i,m}$ $(1 \le a_{i,j} \le n \times m)$ where $a_{i,j}$ indicates the integer in cell (i, j).

It's guaranteed that the sum of $n \times m$ of all test cases will not exceed 10^5.

Output

For each test case, output one line containing one integer indicating the sum of values of all possible paths. As the answer might be large, output it modulo 998 244 353.

Example

standard input	standard output

3
2 3
5 2 1
1 5 5
1 1
1
2 3
3 3 3
3 3 3
	7
	1
	3

Note

For the first sample test case, there are 3 possible paths.

- The first path is $(1, 1) \to (1, 2) \to (1, 3) \to (2, 3)$. $\mathbb{S} = \{1, 2, 5\}$.
- The second path is $(1, 1) \to (1, 2) \to (2, 2) \to (2, 3)$. $\mathbb{S} = \{2, 5\}$.
- The third path is $(1, 1) \to (2, 1) \to (2, 2) \to (2, 3)$. $\mathbb{S} = \{1, 5\}$.

So the answer is $3 + 2 + 2 = 7$.

Problem H. WildFire, This Is for You!

Foammm has created the following problem.

Coprime Problem

You are given two positive integers x and y. The following four operations can be performed with a cost of 1 any number of times:

- Add 1 to x.
- Add 1 to y.
- Subtract 1 from x.
- Subtract 1 from y.

You need to find the minimum total cost to make these two integers have no common divisors greater than 1.

Foammm needs to generate some test cases for the problem. She gives you an integer k, and you need to find two positive integers x and y so that the answer to the above problem is exactly k.

Input

There is only one test case in each test file.

The first and only line contains an integer k ($0 \le k \le 20$).

Output

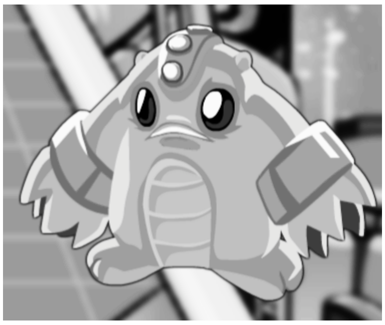
Output an integer x ($0 < x < 10^{1500}$) in the first line, followed by an integer y ($0 < y < 10^{1500}$) in the second line.

It can be shown that the answer always exists. If there are multiple valid answers, you can output any of them.

Examples

standard input	standard output
0	1 1
1	2 2
2	945 1210

Note



Problem I. Bingo 3

In this problem, you will need to construct a grid with n rows and n columns. Each cell of the grid has an integer in it, where $a_{i,j}$ indicates the integer in the cell located at the i-th row and the j-th column. Each integer from 1 to n^2 (both inclusive) appears exactly once in the grid.

We say an integer x is a "bingo integer" of this grid, if at least one of the two following conditions is satisfied.

- There is at least one row, where all integers in the cells of that row are less than or equal to x.
- There is at least one column, where all integers in the cells of that column are less than or equal to x.

It is easy to see that a grid may have multiple bingo integers, however in this problem, we're only interested in the smallest bingo integer.

Given integers n and k, construct a grid with n rows and n columns, such that its smallest bingo integer is exactly k.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 50$) indicating the number of test cases. For each test case:

The first and only line contains two integers n and k ($1 \le n \le 50$, $1 \le k \le n^2$).

Output

For each test case:

- If it is possible to construct a grid with n rows and n columns, such that its smallest bingo integer is k, first output `Yes` in one line. Then output n lines, where the i-th line contains n integers $a_{i,1}, a_{i,2}, \cdots, a_{i,n}$ separated by a space, indicating the integers on the i-th row of the grid. Don't forget that each integer from 1 to n^2 (both inclusive) must appear exactly once in the grid. If there are multiple valid answers, you can output any of them.
- If it is impossible to find an answer, just output `No` in one line.

Example

standard input	standard output

4
3 5
4 10
5 2
1 1
	Yes
	4 2 5
	7 1 9
	8 6 3
	Yes
	14 9 2 13
	1 11 16 8
	10 3 7 5
	6 15 4 12
	No
	Yes
	1

Problem J. Dictionary

You're going to learn words from a dictionary for the next q days. The dictionary is a string $S = s_1 s_2 \cdots s_n$ of length n consisting of lower-cased English letters, and a word is a substring of S.

On the i-th day, you will learn all the words which start with $S[l_i : r_i] = s_{l_i} s_{l_{i+1}} \cdots s_{r_i}$ (that is to say, $S[l_i : r_i]$ is a prefix of these words). After each day, count how many different words you have already learned starting from the first day.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$) indicating the number of test cases. For each test case:

The first line contains a string $s_1 s_2 \cdots s_n$ of length n ($1 \le n \le 2 \times 10^5$) consisting of lower-cased English letters, indicating the dictionary.

The second line contains an integer q ($1 \le q \le 2 \times 10^5$), indicating the number of days you'll learn words.

For the following q lines, the i-th line contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$), indicating that on the i-th day, you will learn all the words with a prefix of $s_{l_i} s_{l_{i+1}} \cdots s_{r_i}$.

It's guaranteed that neither the sum of n nor the sum of q of all test cases will exceed 2×10^5.

Output

For each test case, output one line containing q integers $c_1, c_2, \cdots c_q$ separated by a space, where c_i is the number of different words you have learned from the 1-st day to the i-th day (both inclusive).

Example

standard input	standard output
2 abcabd 3 1 3 1 2 6 6 aaa 3 3 3 2 3 1 3	4 6 7 3 3 3

Note

For the first sample test case:

- On the 1-st day you learned `abc`, `abca`, `abcab`, and `abcabd`. So the answer is 4.
- On the 2-nd day you learned `ab`, `abc`, `abca`, `abcab`, `abcabd`, and `abd`. There are 4 words you have already learned, so the answer is $4 + (6 - 4) = 6$.
- On the 3-rd day you learned `d`, which you haven't learned before. So the answer is $6 + 1 = 7$.

For the second sample test case, you have learned all the words (`a`, `aa`, `aaa`) on the 1-st day, so the answer is always 3.

Problem K. Las Vegas

Las Vegas is an easy-to-learn dice-rolling board game. Players roll dice and commit them to one of the casinos, and the player with the most dice on a casino wins its money, but only if they're not tied! The game perfectly combines luck and strategy, and was nominated for the *Spiel des Jahres* prize in 2012.

Photo by @denislargeron on BoardGameGeek

Consider a game with n casinos and $(m+1)$ players. You're the $(m+1)$-th player, and all other players have already placed (or decided not to place) their dice on the casino. It is known that on the i-th casino, the j-th player has placed $a_{i,j}$ dice on it.

Your task is to decide on the number of dice b_i you want to place on the i-th casino for each $1 \le i \le n$. After you place your dice, we'll determine the winner of each casino separately according to the following rule:

1. First, if two or more players have placed the same number of dice on this casino, all their dice on this casino will be removed.
2. After that, the player with the most number of dice on this casino is determined to be the winner. If there are no dice on this casino, there is no winner.

For example:

- Consider a casino on which 7 players have placed $1, 2, 3, 3, 4, 4, 4$ dice respectively. First, as there are two or more players with 3 or 4 dice, all their dice are removed, and the remaining number of dice are $1, 2, 0, 0, 0, 0, 0$ respectively. The winner of this casino is the player with 2 dice.
- Consider a casino on which 4 players have placed $2, 3, 2, 3$ dice respectively. First, as there are two or more players with 2 or 3 dice, all their dice are removed, and the remaining number of dice are $0, 0, 0, 0$ respectively. As there are no dice on this casino, there is no winner.

Your goal is to win in the most number of casinos among all players. More precisely, let w_i be the number of casinos where player i is the winner, there must be $w_{m+1} \ge w_i$ for all $1 \le i \le m$. Minimize the total number of dice you place in all casinos while achieving your goal. That is, minimize $\sum_{i=1}^{n} b_i$.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 100$) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n, m \le 50$), indicating the number of casinos and the number of players (except you).

For the following n lines, the i-th line contains m integers $a_{i,1}, a_{i,2}, \cdots, a_{i,m}$ ($0 \le a_{i,j} \le 10^8$), where $a_{i,j}$ is the number of dice on casino i placed by player j.

It's guaranteed that there are at most 3 test cases satisfying $n > 20$ or $m > 20$.

Output

For each test case, output one line containing n integers $b_1, b_2, \cdots, b_n$ ($0 \le b_i \le 10^9$) separated by a space, where b_i is the number of dice you place on casino i. It can be proven that an optimal answer always exists under such constraints of b_i. If there are multiple optimal answers, you can output any of them.

Example

standard input	standard output

3
4 3
3 3 2
2 7 5
3 5 2
1 6 4
3 4
100 100 100 1
0 0 0 1
100 100 100 1
1 4
20 0 20 0
	4 0 6 0
	1 0 2
	0

Note

For the first sample test case, you are the winner of casino 1 and 3, while player 2 is the winner of casino 2 and 4.

For the second sample test case, you are the winner of casino 3, while player 4 is the winner of casino 2. There is no winner for casino 1.

Problem L. Subsequence

We say an integer sequence $C = c_1, c_2, \cdots, c_k$ of length k is good if the average value of its smallest element and its largest element is equal to its median. The median of a sequence of length k is defined to be the $\lceil \frac{k}{2} \rceil$-th smallest element in the sequence, where $\lceil x \rceil$ is the smallest integer larger than or equal to x. For example, the median of sequence $1, 7, 4, 3$ is 3, while the median of sequence $5, 8, 2, 1, 6$ is 5.

More formally, let $D = d_1, d_2, \cdots, d_k$ be the sequence obtained by sorting sequence C from small to large. C is good if

$$\frac{d_1 + d_k}{2} = d_{\lceil \frac{k}{2} \rceil}$$

Given an integer sequence $A = a_1, a_2, \cdots, a_n$, calculate the length of its longest good subsequence. Recall that a sequence B is a subsequence of A if B can be obtained by deleting some or no elements from A, without changing the order of the remaining elements.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 300$) indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 3 \times 10^3$), indicating the length of the sequence.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 10^9$), indicating the sequence.

It's guaranteed that the sum of n of all test cases does not exceed 3×10^3.

Output

For each test case, output one line containing one integer, indicating the length of the longest good subsequence.

Example

standard input	standard output

4
7
3 5 9 8 2 11 5
7
7 9 2 4 17 10 15
1
100
2
100 100
	5
	4
	1
	2

Note

For the first sample test case, the longest good subsequence is $3, 5, 8, 2, 5$. It's smallest element is 2, largest element is 8, and median is 5.

For the second sample test case, the longest good subsequence is $7, 9, 4, 10$. It's smallest element is 4, largest element is 10, and median is 7.

Problem M. Flight Tracker

Let the Earth be a sphere centered at $(0, 0, 0)$ with a radius r in 3D Euclidean space. There is a flight flying along the shortest path from the departure place to the destination place on the surface of the Earth.

As an aviation enthusiast, you have a receiver that can receive the signal from the flight with a distance no more than d. Note that we calculate the distance between two points by measuring the shortest path on the surface of the Earth, which is NOT the Euclidean distance in 3D Euclidean space. You need to find the minimum d so that you can receive the signal from the flight at some time with the receiver in your place.

Input

There are multiple test cases. The first line of the input contains an integer T $(1 \le T \le 10^4)$ indicating the number of test cases. For each test case:

The first line contains an integer r $(1 \le r \le 100)$ indicating the radius of the Earth.

The second line contains three integers a, b and c $(-100 \le a, b, c \le 100,\ a^2 + b^2 + c^2 > 0)$, indicating that your place has coordinates $\left(\frac{ra}{\sqrt{a^2+b^2+c^2}}, \frac{rb}{\sqrt{a^2+b^2+c^2}}, \frac{rc}{\sqrt{a^2+b^2+c^2}}\right)$.

The third line contains three integers u, v and w $(-100 \le u, v, w \le 100,\ u^2 + v^2 + w^2 > 0)$, indicating that the departure place has coordinates $\left(\frac{ru}{\sqrt{u^2+v^2+w^2}}, \frac{rv}{\sqrt{u^2+v^2+w^2}}, \frac{rw}{\sqrt{u^2+v^2+w^2}}\right)$.

The fourth line contains three integers x, y and z $(-100 \le x, y, z \le 100,\ x^2 + y^2 + z^2 > 0)$, indicating that the destination place has coordinates $\left(\frac{rx}{\sqrt{x^2+y^2+z^2}}, \frac{ry}{\sqrt{x^2+y^2+z^2}}, \frac{rz}{\sqrt{x^2+y^2+z^2}}\right)$.

It is guaranteed that the departure place and the destination place cannot coincide with each other and cannot be directly opposite each other on the Earth. Therefore, the shortest path from the departure place to the destination place on the surface of the Earth is uniquely determined.

Output

For each test case, output one line containing one real number, indicating the minimum d so that you can receive the signal from the flight at some time with the receiver in your place.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-4}. Formally speaking, suppose that your output is a and the jury's answer is b, and your output is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-4}$.

Example

standard input	standard output

2
100
1 1 1
1 0 0
0 1 0
100
-1 -1 0
1 0 0
0 1 0
	61.547970867038734110
	235.619449019234492887

Note

The following figure illustrates the first sample case, where P is your place, S is the departure place, and T is the destination place.