Problem A. Bits and Tricks

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Blue`

The AI got better—it started giving explanations. But the team didn't read them much. "It works,"they said. But when someone asked, "Why does it work? they had no answer.

Given two integers A, B and a range of numbers $L,\ L+1,\ L+2,\ \ldots,\ R$ $(L \leq R)$.

You can do the following operation any number of times:

- Choose a number x, where $x \in [L, R]$.
- Choose an **on bit** i from x, where i is an index of x's binary representation. Bit i is called **on** in x's binary representation if $x \,\&\, 2^i \;=\; 2^i$.
- Make $B = B|2^i$.

Output `"YES"` if you can make $A = B$ using the above operation any number of times; otherwise, print `"NO"`.

Note: Please read the explanatory note at the end for definitions bitwise AND (&) and bitwise OR (|).

Input

Each test contains multiple test cases. The first line contains the number of test cases T $(1 \leq T \leq 10^4)$. The description of the test cases follows.

The first line of each test case contains four integers $A,\ B,\ L,\ R$ $(0 \leq A, B \leq 10^9)$ $(1 \leq L \leq R \leq 10^9)$.

Output

For each test case, output `"YES"` (without quotes) if you can make $A = B$ using the mentioned operation; otherwise, output `"NO"` (without quotes).

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

Example

standard input	standard output
3 3 1 8 10 3 1 4 5 3 10 1 100	YES NO NO

Note

Bitwise AND (&):

The Bitwise AND ("&") operator compares two numbers on a bit-by-bit basis and returns a new number where a bit is set to 1 only if that bit is 1 in **both** original numbers.

Example: $5 \,\&\, 3$ is 1 because $101_2 \,\&\, 011_2 = 001_2$.

Bitwise OR ("|")

The Bitwise OR ("|") operator compares two numbers bit-by-bit and returns a new number where a bit is set to 1 if that bit is 1 in **either** of the original numbers.

Example: $5|\ 3$ is 7 because $101_2 \mid 011_2 = 111_2$.

Problem B. Triple Adjacent Coprimes

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Silver`

This time, they went back to pen and paper. They spent hours thinking, drawing, and solving step by step. They were slower—but they understood better. They started enjoying problem-solving again.

You are given two positive integer numbers N and M.

You need to count the number of arrays A of length N, which satisfy the following conditions:

- $(1 \le A_i \le M)$ for each i $(1 \le i \le N)$.
- $\gcd(A_{i-1}, A_i, A_{i+1}) = 1$ for each i $(2 \le i \le N-1)$.

Input

The first line contains a single positive integer number $(1 \le T \le 4000)$, the number of test cases.

Each of the following T lines contains two positive integer numbers N, M $(1 \le N \le 4000)$, $(1 \le M \le 400)$.

It is guaranteed that the sum of N over all testcases does not exceed 4000.

And there's no constraints on the sum of M over all testcases.

Output

For each testcase, in a separate line, you need to output the number of arrays A satisfying the conditions in the problem statement. Since the number could be large, output it modulo $10^9 + 7$.

Example

standard input	standard output
20 1 8 2 6 3 20 4 17 5 12 6 11 7 16 8 15 9 5 10 17 11 3 12 10 13 11 14 15 15 14 16 9 17 15 18 9 19 11 20 18	8 36 6745 65949 167376 1202933 141792081 282209474 1249291 361954701 103505 426697206 189791575 672991432 84441627 110860457 563869081 703996889 96277609 88476841

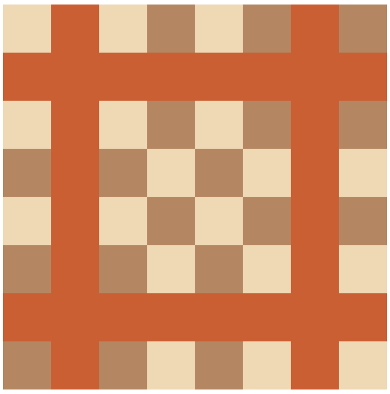
Problem C. Eman's Board

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Yellow`

In the third week, a small bug appeared. No one could find it. They gave the code to the AI, but it didn't help—it didn't understand code they didn't write. They forgot how to debug by themselves.

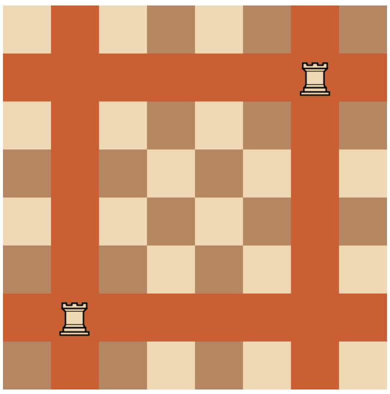
Eman has an $N \times M$ chessboard, where N represents the number of rows and M represents the number of columns. Rocks will be placed on the board such that **all marked squares must be attacked**, and **no unmarked square can be attacked**.

She now seeks to determine the maximum and minimum number of rocks that can be placed on the board to produce the same marked squares. For example, consider the following initial board with marked attack squares:

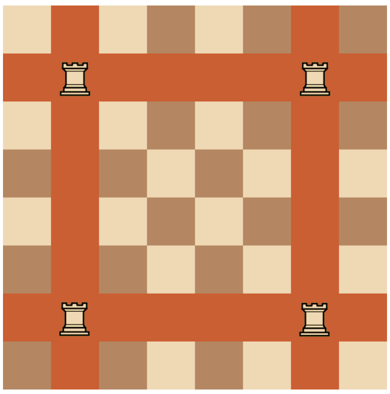


Initial board with marked attack squares

For this board, the maximum number of rocks that can be placed to recreate the same marked squares is 4, while the minimum number is 2. The configurations are shown below:



Configuration with 2 rocks



Configuration with 4 rocks

Input

The first line contains two integers N and M ($1 \leq N, M \leq 10^3$), denoting the dimensions of the chessboard.

The following N lines contain M characters each, representing the initial state of the board. Each character can be either a '#' representing a marked square or a '.' representing an unmarked square.

It is guaranteed that the board is valid — that is, there exists at least one configuration of rocks that can produce the given pattern of marked squares.

Output

Output two integers, separated by a space, representing the maximum and minimum number of rocks that can be placed on the board to make the same marks.

Examples

standard input	standard output
8 8 .#....#. ######## .#....#. .#....#. .#....#. .#....#. ######## .#....#.	4 2
2 2 .# ##	1 1

Problem D. Towering Triumph

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`White`

The team had just discovered a revolutionary AI coding assistant. With a single sentence, it could write entire functions. At first, it felt like magic—an endless stream of working solutions. But none of the team members could explain how the solution worked. Still, they moved on.

During the ECPC Qualification, the team's balloon got stuck on the ceiling! The only way to retrieve it is by stacking their three chairs, and the tallest of the three teammates has to make the climb.

Given their three heights, find the height of the climber.

Input

The first and only line of input contains three space-separated integers: h_1, h_2, h_3 $(150 \leq h_1, h_2, h_3 \leq 210)$ – representing the heights of the three teammates in centimeters.

Output

Print a single integer, which is the height of the tallest teammate.

Example

standard input	standard output
`155 160 180`	`180`

Problem E. Easy Counting!

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Balck`

At the training camp, another team saw someone using AI during a contest. People started asking: is this training or cheating? Trust was broken, and fairness was in danger.

Given two integers L, and R ($L \leq R$). Find the number of different non-empty arrays modulo 10^9+7 such that the sum of its elements is between L and R inclusively and all of its elements are pairwise coprime. i.e., the gcd of any pair of the array is equal to 1.

Input

The first line contains a single integer T — denoting the number of test cases ($1 \leq T \leq 10^5$).

Then T lines follow each containing two integers L, and R ($1 \leq l \leq r \leq 225$).

Output

For each test: print a single integer modulo 10^9+7 — denoting the answer to the problem.

Example

standard input	standard output
2 1 3 4 5	7 20

Problem F. Enta leh d3eef

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Orange`

Even after many hours of practice, they didn't get better. They were not solving problems—they were only watching the AI solve them. They were not learning anything new. Their progress stopped.

You are given two positive integers A and K. Your task is to find the non-negative integer B such that

$$\gcd(A, B) + \operatorname{lcm}(A, B) = K$$

If no such integer B exists, output `-1`.

Input

The first line contains a single integer T ($1 \le T \le 10^5$) — the number of test cases.

Each of the following T lines contains two space-separated integers A and K ($1 \le A \le K \le 10^9$).

Output

For each test case, print a single integer — the value of B or print `-1` if no such B exists. You may print any valid answer if more than one exists.

Example

standard input	standard output
1 6 8	2

Note

- $\gcd(A, B)$ stands for the greatest common divisor of A and B, which is the largest positive integer that divides both A and B.
- $\operatorname{lcm}(A, B)$ stands for the least common multiple of A and B, which is the smallest positive integer that is a multiple of both A and B.

Problem G. Mexy Grid

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Pink`

In practice contests, they did well using AI. They felt confident. But in the real regional contest—where tools were not allowed—they froze. No hints. No code. Only silence.

You are given a grid A with N rows and M columns, containing integers. You are allowed to swap any two elements in the entire grid **at most one time**.

The **value** of the grid is defined as the sum of the **MEX** of each row. The MEX (Minimum EXcluded value) of a row is the smallest non-negative integer that does not appear in that row.

Your task is to find the maximum possible value of the grid that can be achieved by performing **at most one swap**.

Input

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

The first line of each test case contains two integers N and M $(1 \le N \times M \le 3 \times 10^5)$ — the dimensions of the grid.

Then N lines follow, each containing M integers $A_{i,j}$ $(0 \le A_{i,j} \le 10^9)$.

It is guaranteed that the sum of $N \times M$ over all test cases doesn't exceed 3×10^5.

Output

For each test case, print a single integer — the maximum total value of the grid (sum of row MEX values) achievable after performing at most one swap.

Examples

standard input	standard output
2 3 5 0 1 2 3 3 0 1 2 2 2 0 1 2 4 4 3 3 2 1 2 0 0 0 0 1 1	13 6
2 1 1 0 2 5 0 2 2 3 4 0 0 1 3 2	1 6

Problem H. Graph Journey

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Lim Green`

A simple graph problem came up. The AI gave an answer quickly—but it was wrong. No one noticed until the scoreboard didn't change. They had stopped checking answers and lost the habit of proving their code.

You are planning a trip through a city from a starting location (node 1) to a destination (node N). The city map is represented as a weighted, connected, undirected graph.

The weight of an edge is the cost to travel along that road. It's allowed to traverse the same road any number of times, and its cost is added for each traversal.

To make things interesting, certain locations (nodes) offer a special **one-time** travel discount. When you arrive at one of these special locations, its discount value is subtracted from your total travel cost. Each discount can only be collected once, even if you visit the same location multiple times.

Your task is to find the minimum possible cost to travel from node 1 to node N.

It is guaranteed that there is always a path from node 1 to node N.

Input

The first line contains three positive integers N, M, and K ($1 \leq N, M \leq 10^5$ and $1 \leq K \leq 18$), representing the number of nodes, the number of edges, and the number of special nodes, respectively.

Each of the next M lines contains three integers u_i, v_i, and w_i ($1 \leq u_i, v_i \leq N$ and $1 \leq w_i \leq 10^9$). This means there is an edge between nodes u_i and v_i with a value of w_i.

The next K lines each contain two integers u and val ($1 \leq u \leq N$ and $1 \leq val \leq 10^9$), indicating that node u is a special node with a magic value of w.

Output

Print the minimum possible cost to travel from node 1 to node N.

Examples

standard input	standard output
6 8 3 6 2 8 5 2 9 6 4 2 3 4 10 3 5 2 1 4 1 2 1 7 3 6 10 2 5 3 4 5 10	3
10 11 5 7 10 6 8 7 17 9 2 14 6 4 15 5 4 1 1 6 9 9 3 16 7 1 9 9 4 12 9 10 6 6 7 10 1 3 4 8 5 11 9 8 10 16	-4

Problem I. Boring Lecture

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Light Blue`

The AI often gave similar ideas: greedy, then DP, then brute force. The team stopped thinking of new ways. Their creativity became weaker. They just waited for the same old ideas.

In a boring lecture held in a hall represented as a two-dimensional grid of size $N \times M$, each cell contains a seat.

The person sitting in seat (i, j) will get bored after $A_{i,j}$ minutes from the start of the lecture and will want to leave.

However, the person will only leave if two conditions are satisfied:

- At least $B_{i,j}$ people have already left the lecture before them.
- There exists a path from their seat to any edge of the grid such that all seats along the path are empty (i.e., already vacated), allowing the person to walk out.

Once both conditions are satisfied, the person leaves immediately, without consuming any additional time.

Your task is to determine, for each person, at what time he will leave the lecture, or if he will never leave.

Input

The first line contains two integers N and M $(1 \le N \times M \le 3 \times 10^5)$ — the number of rows and columns of the grid.

Then N lines follow, each containing M integers $A_{i,j}$ $(0 \le A_{i,j} \le 10^9)$ — the minute each person starts wanting to leave.

Then N lines follow, each containing M integers $B_{i,j}$ $(0 \le B_{i,j} \le N \times M)$ — the minimum number of people that must have left before the person at (i, j) is willing to go.

Output

Print N lines, each containing M integers. The value at position (i, j) should represent the minute at which the person in that seat leaves the lecture.

If a person never leaves, print `-1` instead.

Examples

standard input	standard output
2 2 1 0 3 4 1 2 0 1	3 3 3 4
4 4 5 0 9 2 9 4 15 7 9 2 7 12 0 15 14 15 12 6 2 16 11 1 1 1 6 5 16 6 0 0 4 2	15 15 9 -1 15 15 15 7 15 15 -1 15 0 15 15 15

Problem J. The Flipping Parade

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Green`

Before using AI, they used to talk a lot, draw, and discuss ideas. Now, everyone was silent, looking at their own screen. They talked less, worked less together. The team spirit started to disappear.

Bob is playing a game with an array A of length N. Initially, all elements of A are zero.

The game proceeds for M minutes. In each minute i (from 1 to M), you are given a position x_i. This means the robot's presence at position x_i is placed:

Let array R be the array of all robots' positions placed so far, sorted in ascending order. $R_1, R_2, \dots, R_i$

- The 1st and 2nd robots form a pair, flipping the segment $[R_1, R_2]$ (inclusive).
- The 2nd and 3rd robots form a pair, flipping the segment $[R_2, R_3]$.
- ...and so on. The i-th pair of robots (at positions R_i and R_{i+1}) flips the segment $[R_i, R_{i+1}]$.

A `flip` on a segment changes all zeros to ones and all ones to zeros.

After exactly M minutes, the game ends. Bob wants to know the total number of ones in the array A based on the final configuration of robots.

Input

The first line contains two integers N and M ($1 \le N, M \le 10^5, M \le N$).

The second line contains M space-separated integers $x_1, x_2, \dots, x_M$ ($1 \le x_i \le N$), representing the position given at each minute.

Output

Print a single integer — the total number of ones in the array A after all M operations are complete.

Example

standard input	standard output
6 4 4 6 3 1	6

Note

Explanation of the first example: Let R be the set of robot positions.

- Initially, $R = \emptyset$.
- Minute 1: $x_1 = 4$. A robot is placed. $R = \{4\}$.
- Minute 2: $x_2 = 6$. A robot is placed. $R = \{4, 6\}$.
  - flip segment $\{4, 6\}$
- Minute 3: $x_3 = 3$. A robot is placed. $R = \{3, 4, 6\}$.
  - flip segments $\{3, 4\}, \{4, 6\}$
- Minute 4: $x_4 = 1$. A robot is placed. $R = \{1, 3, 4, 6\}$.
  - flip segments $\{1, 3\}, \{3, 4\}, \{4, 6\}$

Problem K. Maximum Path Score

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Cyan`

They didn't qualify. Using AI too much has made them weak. "Next year,"one said, "but we won't use shortcuts."They learned that AI is strong, but they had used it the wrong way. It was a hard lesson.

Given a tree with N nodes, numbered from 1 to N. Each node i has an integer weight w_i. A path in the tree is any simple path between two nodes (possibly the same node). If a path uses k edges and the sum of the weights of its nodes is S, then its score is defined as:

$$\textbf{score} = S - k^2.$$

What is the maximum score over all possible paths in the tree, so he can boast to 7oka about his cleverness!

Input

The first line contains a single integer T ($1 \le T \le 10^4$) — the number of test cases.

The description of the test cases follows.

The first line of each test case contains one integer N ($1 \le N \le 2 \times 10^5$) — the number of nodes.

The second line contains N integers $w_1, w_2, \ldots, w_N$ ($-10^9 \le w_i \le 10^9$) — the weights of the nodes.

Each of the next $N - 1$ lines contains two integers u_j, v_j ($1 \le u_j, v_j \le N$), denoting an edge between nodes u_j and v_j.

It is guaranteed that the sum of N over all test cases does not exceed 2×10^5.

Output

For each test case, print one integer: the maximum score achievable by any path in the tree.

Example

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

Problem L. Maximum Valid Subarrays

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

Training became a habit: read the problem, then ask the AI. With time, they stopped thinking. "Why should we try?"one said. "The tool always gives correct code."The others agreed, not realizing they were only copying, not learning.

You are given an array A of N integers and a target value K. Your task is to select the maximum number of contiguous and non-overlapping subarrays from the array A.

For a subarray to be included in your selection, it must contain at least one pair of elements (x, y), at different positions within that subarray, such that $x + y = K$.

The subarrays you select cannot overlap. For example, if you select the subarray $A[i \dots j]$, you cannot select any other subarray that includes an index in the range $[i, j]$. Any elements of the original array that are not part of a selected subarray are ignored.

Input

The first line contains two integers N, K $(2 \le N \le 2 \times 10^5)$, $(1 \le K \le 10^9)$.

The second line contains N integers A_i $(0 \le A_i \le 10^9)$.

Output

Print a single integer representing the maximum number of non-overlapping subarrays you can select.

Examples

standard input	standard output
9 2 1 8 1 1 1 1 0 0 1	3
5 5 1 2 3 4 1	2

Problem M. Once Upon a Common Ancestor

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Bronze`

One week before the contest, they used AI only after trying hard themselves. They saw AI as a tool, not a shortcut. Now they were ready—to think, to solve, and to work as a real team.

In this problem, you are given a rooted tree with N nodes, rooted at node 1. You are also given Q queries. Each query consists of a set of K distinct nodes. For each query, you need to compute the following sum: $\sum_{i=1}^{K}\sum_{j=i}^{K}\mathrm{LCA}(u_i, u_j)$ Where:

- $u_1, u_2, \ldots, u_K$ are the nodes in the set,
- $\mathrm{LCA}(u_i, u_j)$ is the lowest common ancestor of nodes u_i and u_j.

Input

The first line contains an integer N $(1 \le N \le 10^6)$ — the number of nodes in the tree.

Each of the next $N-1$ lines contains two integers u and v $(1 \le u, v \le N)$, describing an undirected edge between node u and node v. These edges form a rooted tree (a connected acyclic graph with $N-1$ edges and a unique path from the root to any node).

The next line contains an integer Q $(1 \le Q \le 10^5)$ — the number of queries.

Each of the next Q lines begins with an integer K $(1 \le K \le N)$, followed by K distinct integers $u_1, u_2, \ldots, u_K$ $(1 \le u_i \le N)$, representing the set of nodes for the query.

It is guaranteed that the total sum of all K over all queries does not exceed 10^6.

Output

For each query, print a single integer — the value of: $\sum_{i=1}^{K}\sum_{j=i}^{K}\mathrm{LCA}(u_i, u_j)$

Examples

standard input	standard output
3 1 2 1 3 3 2 1 2 3 2 1 3 1 3	4 9 3
7 1 2 2 3 1 4 3 5 3 6 4 7 2 3 3 4 7 4 5 3 6 1	20 27

Problem N. Top 10 Market Picks — Your Parts' 2026 Expansion Strategy

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Gold`

Your Parts is a Den VC-backed digital B2B platform transforming the fragmented automotive spare parts market across the Arab world. By connecting garages and workshops to verified parts suppliers through a smart sourcing and logistics platform, Your Parts has rapidly gained traction in several MENA countries.

Heading into 2026, Your Parts plans to expand across Africa and the Middle East. However, successful expansion requires careful market prioritization. Out of n potential countries, the company must select exactly k markets that are the most strategically aligned with its growth plans.

Each market is evaluated using 10 key parameters, such as vehicle fleet size, garage density, logistics infrastructure, digital maturity, ease of market entry, number of Chinese cars, EV growth, tax and import regulations, purchasing power, and population size. Each parameter has an associated weight, which reflects its importance to Your Parts' 2026 strategy.

The total strategic score for a market is the weighted sum of its parameter scores. Your goal is to help Your Parts choose exactly k markets with the highest possible total strategic value.

Input

The first line contains two integers n and k ($1 \leq k \leq n \leq 65, k \leq 10$) — the total number of markets and the number of markets to select.

The second line contains 10 integers $w_1, w_2, \ldots, w_{10}$ ($1 \leq w_i \leq 100$) — the weights for each of the 10 parameters.

Each of the next n lines contains 10 integers $s_1, s_2, \ldots, s_{10}$ ($0 \leq s_i \leq 10^6$) — the scores for each parameter for one market.

Output

Print a single integer — the maximum possible total strategic value by selecting exactly k markets.

Example

standard input	standard output
5 2 1 2 3 4 5 6 7 8 9 10 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5	495

Note

In the sample, the parameter weights are $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$.

A market with scores $[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ has a total score of 55.

A market with all scores 5 has a score of $5 \times 55 = 275$.

The best two markets to pick are those with scores 275 and 220.

The maximum total is $275 + 220 = 495$.