Problem A. Easy Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Askhat is a prospective businessman. He quickly figured that programming is an unprofitable business, so he decided to open a chicken farm.

His farm consists of n chickens ordered in a row. The i-th chicken can eat at most a_i grains. There are m feeders, each described by integers l_j, r_j, c_j. The j-th feeder can feed the i-th chicken if $l_j \le i \le r_j$, and there are c_j grains in this feeder.

Turns out that every business has its own pitfalls, in this case it has the face of chicken feeding control, represented by Ildar. He claims that every respectable chicken farm must have a chicken representative. That is, there must exist a chicken i such that $l_j \le i \le r_j$ holds for every feeder j. All feeders that don't obey this rule must be exterminated.

Now Askhat asks you to find, for each i, what is the maximum number of grains that can be fed to chickens if we leave only feeders that can feed chicken i.

Input

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

The first line of each test case contains two integers n, m ($1 \le n, m \le 10^5$) — the number of chickens and the number of feeders respectively.

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$) — the number of grains that chickens can eat.

Each of the next m lines contains three integers l_j, r_j, c_j ($1 \le l_j \le r_j \le n$, $0 \le c_j \le 10^9$) — description of the j-th feeder.

It is guaranteed that both the sum of n and the sum of m for all test cases do not exceed 10^5.

Output

For each test case, print n integers — the answer to the problem.

Example

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

Problem B. Standard Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

You are given n segments $[l_i, r_i]$ $(1 \le l_i \le r_i \le m)$. Each segment has a weight c_i.

Let us choose a subsequence of segments, from each chosen segment choose an integer and arrange them in the same order as initial segments. By this operation we will get an integer sequence. We say that a subsequence of segments is **good** if we can construct a **nondecreasing** integer subsequence from it.

Let k be the maximum weight of a good subsequence (the sum of weights of all segments in the subsequence). Calculate k and the number of good subsequences of weight k. Since the number of subsequences can be large, calculate it modulo 998 244 353.

Input

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

The first line of each test case contains two integers n, m $(1 \le n, m \le 2 \cdot 10^5)$.

Each of the next n lines contains three integers l_i, r_i, c_i $(1 \le l_i \le r_i \le m,\ 1 \le c_i \le 10^9)$ — description of the i-th segment.

It is guaranteed that both the sum of n and the sum of m for all test cases do not exceed $2 \cdot 10^5$.

Output

For each test case, print two integers — the maximum weight of a good subsequence and the number of good subsequences with maximum weight (the second number modulo 998 244 353).

Example

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

Problem C. Network Transfer

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Maksim, a well-known computer scientist in the field of networking, came up with a new protocol that Ramazan suggested to call `cerr_maksim`.

For simplicity, let us say that there are two computers in the network, and they are connected with a wire of throughput w. Files are being transferred from the first computer to the second computer. Transferring a file of size s takes $\frac{s}{w}$ seconds.

There are n files to be transferred, each has a moment t_i when it starts being transferred, size s_i and priority p_i. If multiple files are being transferred simultaneously, then wire's throughput is divided between transfers proportionally to their priorities.

For each file, calculate the moment when it will reach the second computer.

Input

The first line contains two integers n, w ($1 \le n \le 2 \cdot 10^5$, $1 \le w \le 10^7$) — the number of files and wire's throughput.

Each of the next n lines contains three integers t_i, s_i, p_i ($1 \le t_i \le 10^7$, $1 \le s_i \le 10^7$, $1 \le p_i \le 100$) — start time of the transfer, size and priority.

Output

Print n real numbers, i-th number being the moment when the transfer of i-th file is completed.

Your answers will be considered correct if, for each of them, its absolute or relative error does not exceed 10^{-6}.

Examples

standard input	standard output
2 10 0 100 2 4 200 1	13 30
2 10 30 200 1 10 100 2	50 20

Problem D. Hard Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You are given an integer array $a_1, \ldots, a_n$. A subsegment of even length $a_i, \ldots, a_{i+2m-1}$ is called **good** if $|\max(a_i, \ldots, a_{i+m-1}) - \max(a_{i+m}, \ldots, a_{i+2m-1})| \le k$.

Let us define an integer sequence f as follows:

- $f_1 = 3240$
- $f_2 = 3081$
- $f_3 = 2841$
- $f_4 = 343$
- $f_i = f_{i-1} \cdot 223 + f_{i-2} \cdot 229 + f_{i-3} \cdot f_{i-4} \cdot 239 + 17$ for $i > 4$

Calculate the sum $(a_{i+m-1} + 10) \cdot f_m$ among all good subsegments. Since this number can be large, print it modulo $998\,244\,353$.

Input

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

The first line of each test case contains two integers n, k ($1 \le n \le 5 \cdot 10^5$, $0 \le k \le \min(n, 10)$).

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$).

It is guaranteed that the sum of n for all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, print a single integer — the answer to the problem.

Example

standard input	standard output
3 6 0 3 1 3 1 3 1 8 4 5 8 4 6 5 7 8 5 7 3 2 1 3 2 2 1 3	144768 745933 448953

Problem E. String Strange Sum

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

You are given a string s consisting of lowercase English letters.

Let us consider a segment $[\ell, r]$ such that $2 \le \ell \le r \le |s|$. Let us define $f(\ell, r)$ as the length of the longest suffix of substring $s[1, \ell - 1]$ such that this suffix can be divided into prefixes of substring $s[\ell, r]$. If there are no such suffixes then $f(\ell, r) = 0$.

Find the sum $\sum\limits_{\ell=2}^{|s|} \sum\limits_{r=\ell}^{|s|} f(\ell, r)$.

Input

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

The only line for each test case contains the string s ($2 \le |s| \le 2 \cdot 10^5$) consisting of lowercase English letters.

It is guaranteed that the sum of $|s|$ for all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the answer to the problem.

Example

standard input	standard output
8 aa ab ababa abaaba abacaba abaaababaab aababcabcbc abcdabcabaabcd	1 0 6 7 0 74 51 20

Note

Let us consider the third test case. In this case, $f(2,2) = 0$, $f(2,3) = 0$, $f(2,4) = 0$, $f(2,5) = 0$, $f(3,3) = 0$, $f(3,4) = 2$, $f(3,5) = 2$, $f(4,4) = 0$, $f(4,5) = 2$, $f(5,5) = 0$. So the answer is 6.

Problem F. Bayan Testing

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Let us recall a well-known problem (also called a "bayan" in Russian). You are given an array $a_1, a_2, \ldots, a_n$ of integers. Answer the queries: given a segment $[l, r]$ $(1 \le l \le r \le n)$, check if there exist two equal elements among $a_l, a_{l+1}, \ldots, a_r$.

Please help to make good tests for this well-known problem! You are given two integers n, m, and also $2m$ **different** segments $[l_i, r_i]$. Find any array $a_1, a_2, \ldots, a_n$ such that, for exactly m queries, the answer is positive, and for exactly m queries, the answer is negative. You should report if there is no such array.

Input

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

The first line of each test case contains two integers n, m $(2 \le n \le 2 \cdot 10^5, 1 \le m \le 10^5)$.

Each of the next $2m$ lines contains two integers l_i, r_i $(1 \le l_i \le r_i \le n)$ — the given segments. It is guaranteed that all segments are different.

It is guaranteed that the sum of n for all test cases does not exceed $2 \cdot 10^5$ and the sum of m for all test cases does not exceed 10^5.

Output

For each test case, print the answer to the problem.

If such an array a exists, print n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$. Otherwise, print a single integer -1.

If there are several possible answers, print any one of them.

Example

standard input	standard output
3 2 1 1 1 2 2 6 2 1 3 4 6 2 4 3 5 4 3 1 2 1 1 2 2 2 3 3 3 3 4	-1 1 2 3 3 2 1 5 5 5 5

Problem G. Battleship: New Rules

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

This is an interactive problem.

Ivan came up with new rules for the battleship game!

- The game will be played on an $n \times n$ board.
- The first player chooses an integer k ($n \le k \le \lceil \frac{n}{2} \rceil^2$).
- After that, the first player places k ships on the board so that the number of cells occupied by the ships is the maximum possible (among all valid placements of k ships of any sizes).
- Each ship should be a rectangle of size $1 \times a$ or $a \times 1$ (a is any integer from 1 to n inclusive). Any two ships should not have neighbouring cells (by side or by corner).

After that, the second player starts his game.

- The second player knows only the size of the board n.
- The second player can ask a query: is cell (x, y) occupied by some ship?
- The second player should find any empty 2×2 square on the board, or say that there are no such squares.

The second player can ask at most $6n$ queries. Please play as the second player and win the game!

Interaction Protocol

The first line contains a single integer t ($1 \le t \le 100$) — the number of games to be played. You should play t games and finish interaction after that.

At the start of the game, you are given a single integer n ($3 \le n \le 1000$) — the size of the board.

After that, you can ask some queries. To ask a query, print a single line "`?` x y" ($1 \le x, y \le n$) — the coordinates of the cell. You will be given an answer c:

- If $c = -1$, you made too many queries. You should terminate your program.
- If $c = 0$, the cell (x, y) is empty.
- If $c = 1$, the cell (x, y) is occupied by some ship.

To finish the game, print a single line "`!` x y", where:

- $x = -1$, $y = -1$ if there are no empty 2×2 squares on the board.
- Otherwise, $1 \le x, y \le n - 1$ and the square with cells (x, y), $(x + 1, y)$, $(x, y + 1)$, $(x + 1, y + 1)$ is empty.

If your answer is incorrect, you will be given a line with value `-1`, and you should terminate your program. Otherwise, you will be given a line with value `1`, and you should play the next game (or finish your program if it was the last game).

It is guaranteed that the sum of n for all games does not exceed 5000.

It is guaranteed that the board in each game is fixed, and the interactor is not adaptive.

Your solution will get `Idleness Limit Exceeded` if you don't print anything or forget to flush the output.

Example

standard input	standard output
2	
3	
	? 2 1
0	
	! -1 -1
1	
4	
	? 1 3
0	
	? 4 3
1	
	! 2 2
1	

Note

Boards from the first test are shown on pictures below. Rows correspond to x coordinates, columns correspond to y coordinates.

	1	2	3
1	1	1	1
2	0	0	0
3	1	0	1

Board from the first game.

	1	2	3
1	1	1	1
2	0	0	0
3	1	0	1

Board from the second game.

In the first game, there are no empty 2×2 squares on the board.

In the second game, there is exactly one empty 2×2 square on the board.

Problem H. Triangular Cactus Paths

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

You are given a simple connected undirected graph that is a cactus: each edge lies on at most one simple cycle. This cactus is triangular: the length of any simple cycle is at most 3.

Answer the queries. In each query, you are given two vertices s and f, and an integer k. Find the number of simple paths between vertices s and f with length exactly k. You should find this number modulo 998 244 353.

The path is simple if all its vertices are different, the length of the path is equal to the number of edges on the path.

Input

The first line contains two integers n, m ($2 \le n \le 2 \cdot 10^5$, $n - 1 \le m \le \frac{3(n-1)}{2}$) — the number of vertices and edges in the graph.

Each of the next m lines contains two integers u, v ($1 \le u, v \le n$, $u \ne v$), meaning that there is an undirected edge (u, v) in the graph. All edges are different. It is guaranteed that the graph is a connected triangular cactus.

The next line contains a single integer q ($1 \le q \le 2 \cdot 10^5$) — the number of queries.

Each of the next q lines contains three integers s, f, k ($1 \le s, f \le n$, $0 \le k < n$) — the description of a query.

Output

Print q integers — the answers to the queries.

Example

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

Problem I. Best Sun

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Ivan likes painting. He decided to paint a sun.

To do that, he took n points with integer coordinates on the plane. Ivan will draw segments connecting some pairs of points to get the best sun.

- Ivan will connect exactly n pairs of points with segments between them.
- All segments should not intersect (except for endpoints).
- There should be exactly one cycle. This cycle should be a convex polygon.
- Each point that is not one of the polygon vertices should lie outside of the polygon and should be connected with one of the polygon's vertices.
- It is possible that all vertices will lie on the cycle.

Ivan wants to paint a bright, pretty sun. So he came up with the score of the sun:

- Let us define S as the area of the polygon.
- Let us define P as the sum of lengths of all drawn segments.
- The value $\frac{S}{P}$ is the score of the sun.

What is the **maximum** possible score of the sun?

Input

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

The first line of each test case contains a single integer n ($3 \le n \le 300$) — the number of points.

Each of the next n lines contains two integers x_i, y_i ($|x_i|, |y_i| \le 10^6$). All points are different. No three points lie on the same line.

It is guaranteed that the sum of n^2 for all test cases does not exceed 90 000.

Output

For each test case, print a single real number — the maximum possible score of the sun that can be drawn.

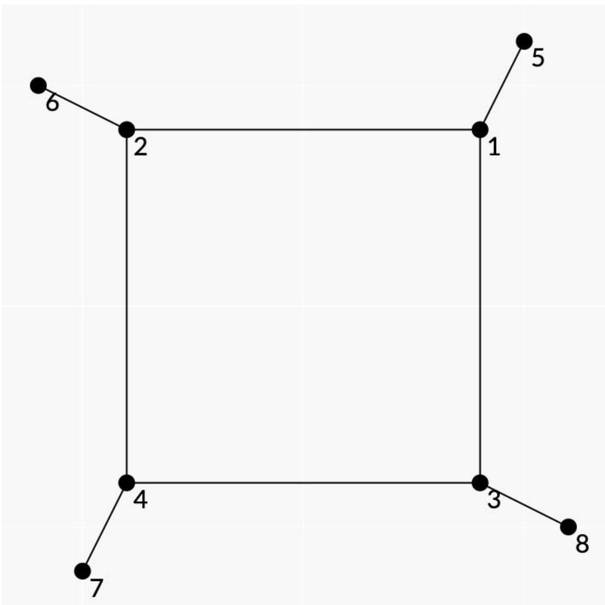
The absolute or relative error should not exceed 10^{-6}.

Example

standard input	standard output
4 3 -1 -1 1 -1 0 1 4 0 0 10 0 0 10 8 1 5 2 0 -2 0 1 1 -1 1 0 3 8 4 4 -4 4 4 -4 -4 -4 5 6 -6 5 -5 -6 6 -5	0.3090169943749474 1.2368614277111258 0.2711375415034555 1.5631002094915825

Note

The picture of the sun with the maximum score in the fourth test case:



For this sun, $S = 64$, $P = 32 + 4\sqrt{5}$, so its score is $\dfrac{64}{32 + 4\sqrt{5}}$.

Problem J. Fast Bridges

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Let us consider a square city of size $k \times k$. There is exactly one house in each cell.

People can go from any cell to neighbouring cell (only by side) in 1 unit of time.

Government decided to build n fast bridges to make the city better. Each fast bridge connects two cells (x_1, y_1) and (x_2, y_2) such that $x_1 \neq x_2$ and $y_1 \neq y_2$. People can go from one end of the bridge to another in $|x_1 - x_2| + |y_1 - y_2| - 1$ units of time.

To analyze how the city became faster, you are asked to calculate the sum of shortest distances between all pairs of cells. Since it can be large, find it modulo 998 244 353.

Input

The first line contains two integers n, k ($0 \leq n \leq 500$, $2 \leq k \leq 10^9$) — the number of bridges and the size of the city.

Each of the next n lines contains four integers x_1, y_1, x_2, y_2 ($1 \leq x_1 < x_2 \leq k$, $1 \leq y_1, y_2 \leq k$, $y_1 \neq y_2$). It is guaranteed that all tuples (x_1, y_1, x_2, y_2) are different.

Output

Print a single integer — the answer to the problem.

Examples

standard input	standard output
2 2 1 1 2 2 1 2 2 1	6
0 1000000000	916520226
5 5 1 1 3 3 3 3 5 1 3 3 4 5 3 3 5 4 1 5 3 3	946

Note

In the first example, the shortest distance between all pairs of cells is 1, so the sum is 6.

Problem K. Decoding The Message

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

Aliens connected with people and sent a message containing the answer to "The Ultimate Question of Life, the Universe, and Everything".

People received n bytes (integers from 0 to 255 inclusive). The decoding algorithm is the following:

- Let us consider all $n!$ permutations of received bytes.
- Consider each permutation as a number written in base 256. Numbers can be equal.
- Multiply all these numbers modulo 65 535.
- The result is the decoded message!

For each byte i, you are given the number c_i of received bytes i. Please decode the message.

Input

The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. Description of test cases follows.

The first line of each test case contains a single integer k ($1 \le k \le 256$) — the number of bytes i such that $c_i \neq 0$.

Each of the next k lines contains two integers i, c_i ($0 \le i \le 255$, $1 \le c_i \le 10^9$). It is guaranteed that all given values i are different.

For all other $256 - k$ bytes, the numbers c_i are equal to 0.

It is guaranteed that $\sum\limits_{i=0}^{255} c_i = n \le 10^9$.

Output

For each test case, print a single integer — the decoded message.

Example

standard input	standard output
5 1 42 1 2 0 1 1 1 1 239 2 2 1 1 2 1 3 1 1 2 2 3 2	42 256 514 1284 61726