Problem A. Autocomplete

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

Peter Mount and Ryei are in a very close Scrabble match. In the game, they must form words and earn points according to the rules. However, the Scrabble played in the MaratonaCIn is a little different: in this version, only words from a specific predetermined set can be formed.

Ryei realized he wouldn't be able to beat Peter Mount, as he is a master of the game. So, he decided to change his strategy: before trying to play a word, he would test which ones were more promising, that is, those that could be transformed into other words within the allowed set. To do this, given a word, he needs to know how many words, that are in the set, he can create, using the given word as a prefix.

Since Ryei is focused on the game, he asked for your help to create a program to assist him in this task. Can you help him?

Input

- The first line contains two integers, N and Q ($1 \leq N, Q \leq 10^5$).
- The next N lines contain one word each, formed only by lowercase letters of the English alphabet.
- The following Q lines contain one word each, formed only by lowercase letters of the English alphabet.
- The sum of the lengths of all the provided words does not exceed 10^6.

Output

- For each of the Q queries, print a single integer on a new line, representing the number of words from the original list that have the query word as a prefix.

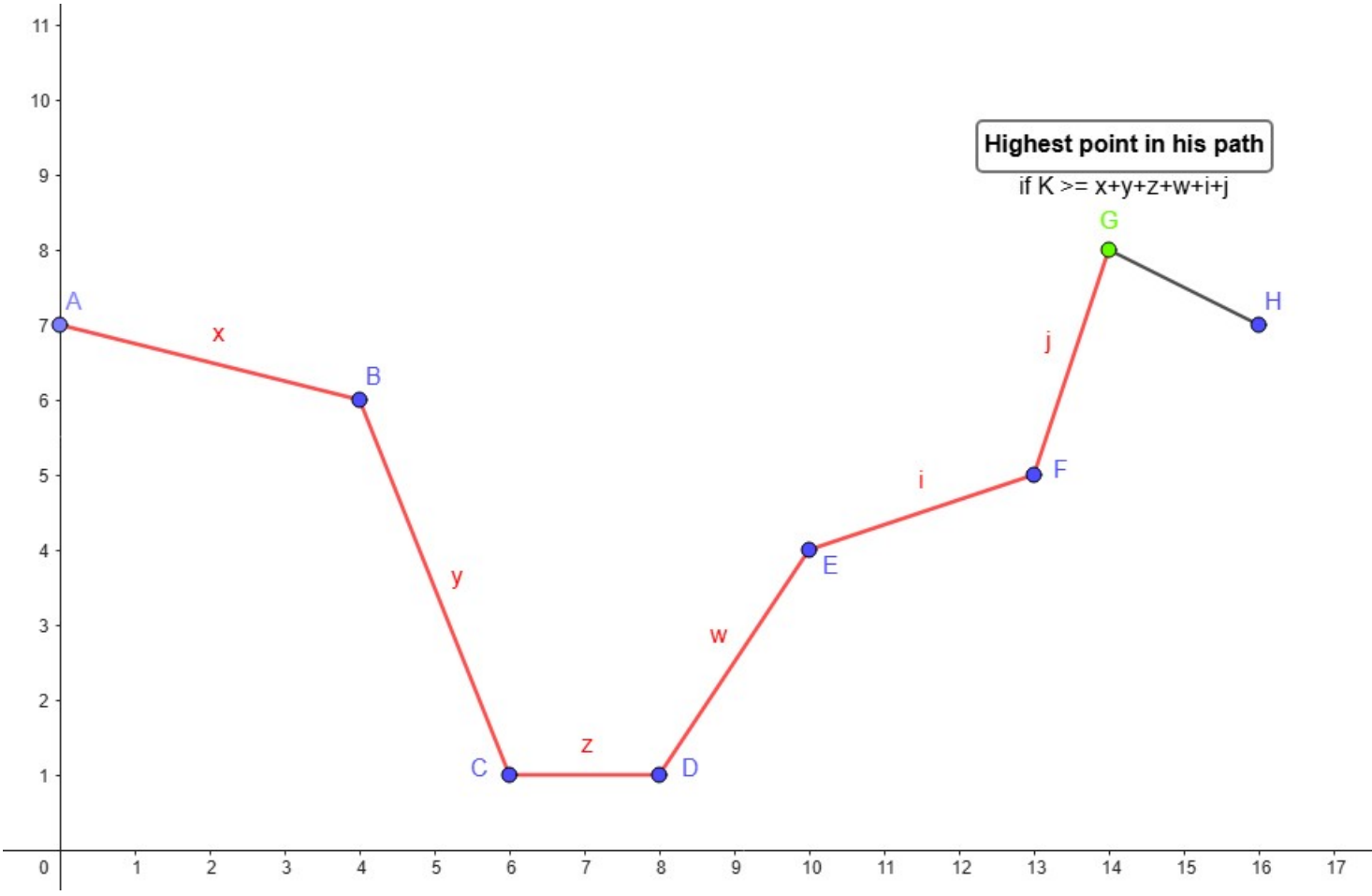
Example

standard input	standard output
4 5 a ab abc abcd a aa ab abc abcd	4 0 3 2 1

Problem B. Beautiful Handsome's Canteen

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

Balloon was going to eat lunch at summer school in Handsome's canteen, and his friends Icarus and Sam Well left him alone on this journey. Unfortunately, the path to Handsome's canteen was steep, and he had to go up and down through many hills.



However, Balloon got tired after walking K meters and fainted from the heat. When he lay down, he remembered how good his life was and how high he had to go. You're given the **sequence of points** in the path to Handsome's canteen, in which the first point describes the location Balloon starts, and your task is to answer how high in the y axis he has gone before fainting.

Input

The input consists of two lines:

The first line contains two integers n, K ($2 \leq n \leq 2 \cdot 10^5$, $0 \leq K \leq 10^{18}$), the number of points and the distance Balloon walked, respectively, in meters.

Then n lines follow. The i-th line contains two integers describing the point p_i: x_i, y_i ($0 \leq x_i, y_i \leq 10^9$), and it is guaranteed they are ordered by x, and no two points have the same x coordinate; that means $x_i < x_{i+1}$ for $1 \leq i < n$.

Output

The output must contain a single line with the real number H, the maximum height he reached. The printed number should have absolute or relative error not exceeding 10^{-4}.

Examples

standard input	standard output
8 22 0 7 4 6 6 1 8 1 10 4 13 5 14 8 16 7	8.000000
8 19 0 7 4 6 6 1 8 1 10 4 13 5 14 8 16 7	7.000000

Note

The path is determined as the sequence of straight segments between every two adjacent points.

The height of a point

$$p_i = (x_i, y_i)$$

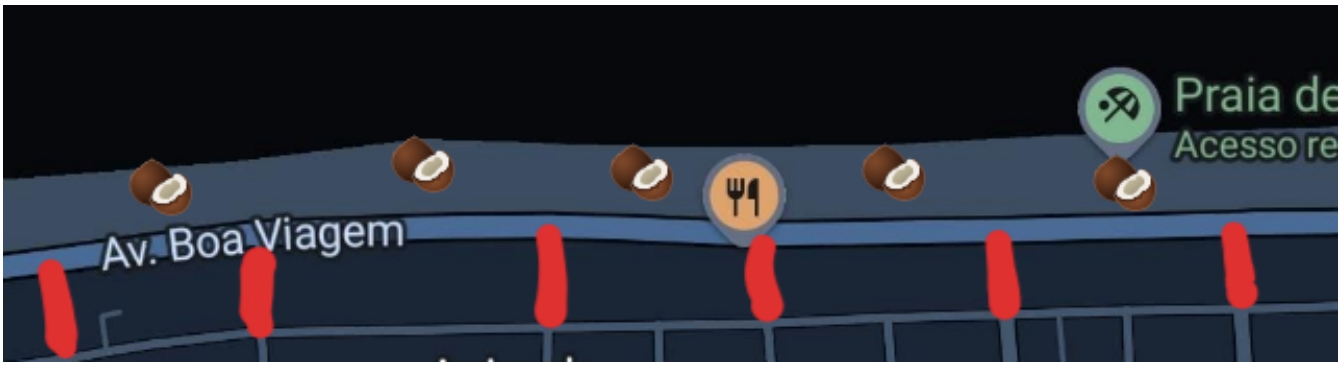
is described by y_i.

Both sample tests are illustrated in the image: the path from A to G is approximately $21.4m$, so for $K = 22m$, he'll go past the point G, achieving the maximum height of 8.

Problem C. Coconuts

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

Leo Angels loves coconuts. He wants to drink coconut water in Good Trip beach. Between every block in Good Trip avenue there is a kiosk with one drink which could be coconut water. Coconuts can taste better than others, being sweeter or colder, each of them having its corresponding satisfaction value. Every block has a street parallel to the avenue, in which Leo could enter or exit. He wants to enter the avenue **once** at any street and get every drink available until his exit.



Leo starts the day with satisfaction $s = 1$ and every time he gets a drink with satisfaction d, his new satisfaction becomes $s \times d$.

Oel knows Leo loves coconuts and changed some of them to Leopsi Zero and sparkling water. Leo hates sparkling water and it always has a negative satisfaction value. However, if his current satisfaction is negative, drinking sparkling water makes it positive again, because life can't be as bad as sparkling water.

When Leo drinks Leopsi Zero, he passes out and his satisfaction becomes zero, because although it is zero calories it doesn't have zero sodium.

Every drink has a satistaction value in the form $|2^k|$ except for Leopsi Zero which is 0.

Find the greatest satisfaction he can get in a day modulo $10^9 + 7$.

Input

The input consists of two lines.

The first line is an integer n ($1 \leq n \leq 2 \cdot 10^5$), the number of blocks in the shore.

The second line contains n integers describing the satisfaction value of each kiosk's available drink $a_1, a_2, ..., a_n$ ($|a_i| <= 2^{30}$) and $|a_i|$ is either 0 or in the form $\pm 2^k$.

Output

You must output one line:

The first and only line contains an integer, the greatest satisfaction value modulo $10^9 + 7$

Examples

standard input	standard output
3 0 -1 0	1
6 8 -1 -2 0 2 4	16
6 8 1 2 0 8 2	16

Note

In the first sample, Leo can choose not to go to the beach and keep his satisfaction at 1.

In the second sample, the greatest satisfaction is 16, and can be achieved by entering in the first block and leaving after third block.

Problem D. Darts

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes



TFG loves to play BloonsTD6 while he's not solving problems in CodeForces. Unfortunately, he is tired of mindlessly popping balloons. He wants to know the exact number of darts required to win the great balloons war challenge. He already knows that for each round he plays, there are n hordes of balloons. In the first horde, there is one balloon; in the second horde, there are two balloons; and so on until the nth horde, where there are n balloons. To defeat a horde with i balloons, TFG has to shoot i^k darts, where k is the balloons' power level.

TFG quickly realized that doing all the math would be too boring for him. Luckily, his friend Lucas figured out that by knowing the number of darts needed to win a balloon battle with power level k, you can calculate the number of darts needed for power level $k+1$ using the formula:

$$S_k = \frac{1}{k+1}\left((1+n)^{k+1} - n - 1 - \sum_{i=1}^{k-1}\binom{k+1}{i}S_i\right)$$

Unfortunately, Lucas is too busy training for the next ICPC stage to calculate the actual number for TFG, so he has asked for your help.

Please help TFG figure out how many darts he needs to end the war.

Since the answer can be large, print it modulo $10^9 + 7$.

Input

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

The only line of each test case contains two integers $n (1 \le n \le 10^9)$ — the number of hordes in the battle and $k (1 \le k \le 10^3)$ — the balloons power level.

Output

For each test case, output a single line with the integer S_k modulo $10^9 + 7$, the number of darts required to win the level.

Examples

standard input	standard output
1 10 1	55
5 10 2 10 3 100 2 8 3 5 5	385 3025 338350 1296 4425

Problem E. Elevator

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

Pep is late for his competitive programming lecture. His building has $n + 1$ floors, and he lives on the floor $n+1$. To go to college, he calls the elevator to his floor. Since he's superstitious, if the elevator stops on more than one floor besides the one it was on when he called and his floor, $n + 1$, he believes the day will be bad and goes back to bed.



While the elevator is moving, each floor i has a probability p_i that the neighbor on floor i will call the elevator at that moment, causing it to stop. Fortunately for Pep, his neighbors are not obsessed with constantly calling the elevator, that is, $p_i < 100\%$ for $1 \leq i \leq n$.

During the college semester, his neighbors may move out, and new neighbors may arrive, changing the probability p_i of floor i. Pep would also like to make queries to determine the probability of going to class, given that the elevator would be on floor x when he calls it.

Input

The first line consists of two integers n ($3 \leq n \leq 2 \cdot 10^5$) and q ($1 \leq q \leq 2 \cdot 10^5$).

The second line consists of n integers p_i ($0 \leq p_i < 100$), representing the probability (in percentage) that the elevator will stop on floor i while in motion.

The next q lines correspond to queries of two types:

- 1 x p: Change the probability of floor x, updating p_x to p.
- 2 x: Answer with the probability that Pep will go to class, given that the elevator is on floor x when he calls it.

Output

For each query of type 2, output a line with the queried probability. We can prove that this probability can be expressed as an irreducible fraction $\frac{p}{q}$. Print $p \cdot q^{-1}$ modulo $(10^9 + 7)$.

Examples

standard input	standard output
3 1 10 50 50 2 1	750000006
5 1 25 25 0 0 0 2 2	1
4 3 40 50 99 99 2 2 1 3 0 2 2	854300006 1

Note

In Sample 1:

The probability of not stopping on floors 2 and 3 is $\frac{1}{4}$.

The probability of stopping only on floor 2 is $\frac{1}{4}$.

The probability of stopping only on floor 3 is $\frac{1}{4}$.

Thus, the final answer is $\frac{3}{4}$.

Problem F. Four is too much

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

The marathonists went to the summer school in Campinas. Now they need to get to Unicamp to participate in the contest. They decided to use a ride-hailing app, Gabmobile. In this app, the user selects the departure and destination locations and has access to ride options from multiple drivers. Each driver asks for a certain price and will only take a certain maximum number of passengers.

The marathonists want to spend as little as possible to go to Unicamp, but they are too tired from training every day to find out the best way to choose the ride options. So they need your help. Given that X marathonists went to the summer school, there are N available ride options and that the i-th ride options is offered by a driver which only wants to take a maximum of p_i passengers and asks for v_i reals for the trip to Unicamp, what is the minimum possible total value spent by the marathonists to get to Unicamp?

Input

The first line of input contains two integers, X and N ($0 < X, N \leq 10^3$), the number of marathonists in the summer school and the number of available ride options, respectively. Following, there are N lines. The i-th one contains two integers, p_i ($0 < p_i \leq 4$) and v_i ($0 < v_i \leq 10^9$), the number of passengers the i-th driver will take and how much they charge for the trip, respectively.

Output

Output an integer Y, the minimum possible total value spent by the marathonists to go to Unicamp. If there is no way for them to get there, output -1.

Examples

standard input	standard output
6 3 3 10 4 20 3 8	18
6 2 3 10 2 5	-1

Problem G. Grisi Maps

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

The marathonists want to walk to the IC at Unicamp, starting from some other place in Campinas. Instead of using GPS, they rely on Grisi, who claims to know how to get anywhere. Grisi always follows the same strategy to reach any destination: from the main entrance of Unicamp, he knows the best way to reach any location in the neighborhood (and vice versa—he knows the best way to get from any location in the neighborhood to the main entrance). So, he always first goes to the main entrance and then to the destination (using the optimal path for both segments).

Given the graph (undirected and connected) representing the neighborhood, the vertex corresponding to the main entrance of Unicamp, the vertex corresponding to the IC, and multiple queries of the form x, representing that the marathonists want to go from vertex x to the IC, determine for each query how much extra time they will take to reach the IC from x by following Grisi's strategy instead of using GPS (which would provide the shortest possible path).

Input

The first line of the input consists of five integers n, m, $q(2 \leq n, m \leq 2 \cdot 10^5, 1 \leq q \leq 2 \cdot 10^5)$, X_1, $X_2(1 \leq X_1, X_2 \leq n)$, representing the number of vertices, edges, queries, the vertex of Unicamp's main entrance, and the IC at Unicamp, respectively.

The next m lines each consist of three integers u, v, $w(1 \leq u, v \leq n, 1 \leq w \leq 1 \cdot 10^9)$, representing an edge connecting vertices u and v with a cost of w time units.

Each of the last q lines consists of a single integer x, representing the vertex from which the marathonists will depart.

Output

The output consists of q lines, where the i-th line indicates the extra time, compared to the GPS route, that the marathonists took to travel from vertex x to the IC at Unicamp following Grisi's method. If the marathonists are already at the IC, the output should be 0.

Example

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

Problem H. Homo Programmius

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



You recently passed the starters tryouts, tasted the pizza of champions after the final contest, and now proudly parade on campus with your ICPC T-shirt every day, monday through saturday (you wash it on sundays).

Your name is now among such legends as Leo Cupids, Gab the Handsome and Big Paul, congratulations!!!

But alas, it seems the times for celebration have come to an end.

You have begun to notice worrisome changes in your physiology.

Symptoms include shrimp posture when coding, social ineptitude and sleepless nights - trying to figure out the proof to a greedy solution.

Alarmed, you take a genetic exam and gasp at the results: doctors have identified sections of your DNA that are unlike any found in a normal human. You are now the world's first *Homo Programmius!*

Your task is to determine how far the mutations have developed.

Specifically, given a DNA string of size n, count how many contiguous segments contain ***at least one*** *mutation. A mutated gene is marked as* **1, while normal genes are written as** 0**.**

For example, the DNA string [001] **has 6 distinct segments:** [(0)01], [0(0)1], [00(1)], [(00)1], [0(01)], [(001)]**. Only 3 of which contain at least one mutation:** [00(1)], [0(01)] **and** [(001)]**.**

But beware! The mutations are unstable, such that any single position of your DNA might change from one instant to the next.

Input

The first line contains an integer n**: the size of the DNA string** ($1 \leq n \leq 10^5$)**.**

The second line consists of a binary string S **of size** n**: the original DNA string** ($S_i = '0'$ *or* $S_i = '1'$)**.**

The third line consists of an integer m**: the number of modifications the DNA string will go through** ($1 \leq m \leq 10^5$)**.**

Each one of the next m **lines corresponds to a modification of the DNA string, consisting of an integer and a character:** pos val**, meaning that** S_{pos} **changes its value to** val**.** ($1 \leq pos \leq n$, $val = '0'$ *or* $val = '1'$)**.**

Output

For each of the m **modification events, a single line-separated integer** x_i **must be printed, such that** x_i **is the number of segments with at least one mutation in the string, after the first** i **modifications**($1 \leq i \leq m$)**.**

Example

standard input	standard output
5 01111 10 4 0 3 0 3 1 2 0 5 0 4 1 4 0 3 0 5 1 1 1	13 11 13 11 9 11 9 0 5 9

Problem I. Intense Duel

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

Samuell and Lleumas are competing to see who can pick the most coconuts. They take turns climbing the coconut tree to pick the coconuts. First, Samuell climbed and picked x coconuts. Then Lleumas climbed and picked y coconuts. From there, the duel intensified. Samuell decided that he would always pick the same number of coconuts that he picked on his last climb, plus the number that Lleumas picked the last time he climbed. Likewise, Lleumas decided to always pick the same number of coconuts that he picked on his last climb, plus the number that Samuell picked the last time he climbed.

Considering the strategies adopted, in the n-th climb, which competitor climbed the coconut tree and with how many coconuts did he climb down modulo $10^9 + 7$?

Input

The input consists of two lines:

The first contains an integer n ($1 \leq n \leq 10^{18}$) - the n-th climb.

The second contains two integers x and y ($0 \leq x, y \leq 10^6$) - the number of initial coconuts that Samuell and Lleumas picked, respectively.

Output

The output must have two lines:

The first must contain a string with the name of the person who climbed the n-th climb.

The second must contain an integer with the number of coconuts picked up in the n-th climb modulo $10^9 + 7$.

Examples

standard input	standard output
3 2 5	Samuell 7
2 3 0	Lleumas 0

Problem J. Judge Fail

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

During a programming contest, the contestants face an unexpected problem: the judging system (judge) has failed and can no longer evaluate the submissions correctly. However, the contest website had prepared for this eventuality by implementing an alternative judge. This alternative judge, for each submission, randomly selects one of the possible verdicts with equal probability.

However, each contestant has restrictions regarding the verdicts they can receive. For example, competitor Gabmei can never receive the "Wrong Answer"(WA) verdict. Thus, for each submission, the system randomly selects, with equal probability, among the verdicts allowed for that particular contestant.

Given that there are **M** contestants in the competition—each with their own list of possible verdicts (from among the **K** available verdicts)—and that **N** submissions have been made (knowing which contestant made each submission), determine the expected frequency of each verdict at the end of the contest.

Input

The first line of the input contains three integers:

- **N** — the number of submissions made ($1 \leq N \leq 100000$);
- **M** — the number of contestants competing ($1 \leq M \leq N$);
- **K** — the total number of possible verdicts ($1 \leq K \leq 10$).

Then, **2 * M** lines are provided regarding the contestants.

For each of the **M** competitors:

- The first line contains the number of verdicts they can receive ($1 \leq qt \leq K$);
- The second line lists, in order, the allowed verdicts for that competitor, represented by distinct numbers **x** ($1 \leq x \leq K$).

Finally, **N** lines follow, each containing the index of the contestant who made the **i**-th submission ($1 \leq i \leq N$).

Output

Output a vector of size **K**, where the **i-th** element epresents the expected frequency of verdict **i** ao final da competição. Cat the end of the contest. Each value must be displayed with at least **6 decimal places**.

Examples

standard input	standard output
4 1 4 4 1 2 3 4 1 1 1 1	1.000000 1.000000 1.000000 1.000000
2 2 5 3 1 2 3 3 1 2 3 1 2	0.666667 0.666667 0.666667 0.000000 0.000000

Problem K. Known Issue

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

Oel's dishonesty is a known issue.

This time he is running for mayor, and, to assure his victory, he's going to hang a poster containing fake news on the greatest wall in the city.

Fortunately, our hero Leo placed some nails on the wall to stop Oel from hanging a big poster. Help Leo find out the area of the biggest poster Oel can hang on the wall.

A poster is a rectangular item that can only be hung on a contiguous nail-free area of the wall, with vertical/horizontal orientation.

Input

The first line of input contains two integers n, m ($1 < n, m < 250$), the dimensions of the wall. The next n lines contain m characters each. The coordinate (i, j) on the wall contains a nail if the j-th character of the i-th line is a '.', and it doesn't if the character is a '#'.

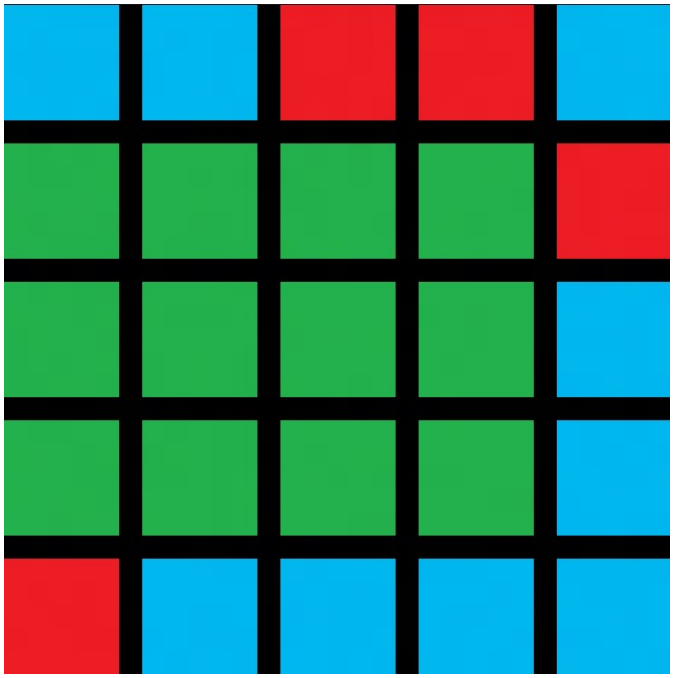
Output

Print a single integer, the area of the biggest poster that can be hung under the given restrictions.

Examples

standard input	standard output
5 5 ##### ##### ##### ##### #####	25
5 5 ##..# ####. ##### ##### .####	12

Note



The green rectangle shows a possible solution for sample 2.

Problem L. Legendary Paper Cup

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Geovany runs the great **Paper Cup** factory! At the factory, Paper Cups of various sizes are produced. More precisely, Paper Cups are produced in all sizes that are an integer x such that $1 \leq x \leq 10^{16}$. These cups are so precisely manufactured that a cup of size x fits perfectly inside a cup of size $x + 1$.



Some sizes of Paper Cups.

But the most important product of the Paper Cup factory is the **Legendary Paper Cup!** A Legendary Paper Cup is produced **by combining** 3 **Paper Cups of sizes** x**,** $(x+1)$**, and** $(x+2)$, and its value becomes $x \times (x+1) \times (x+2)$.

One day, however, Geovany's loyal employees **Henrique** and **Ballon** discovered **counterfeit** Legendary Paper Cups!!! Cheap copies!!! Seeking help from the hero **Leo Angels** to investigate the case, they found out that **Oel Demons** was behind it all!

Oel Demons is advertising that they sell Legendary Paper Cups with all values in a given interval $[L, R]$, but this is obviously a lie, given the manufacturing constraints of Legendary Paper Cups!!!

Now, it's up to you to help our heroes expose Oel Demons' fraud by telling: **how many Legendary Paper Cups in this interval can actually be real?**

More precisely, given several intervals $[L, R]$, determine how many integers $L \leq x \leq R$ are the product of three consecutive integers.

Input

The first line of input contains an integer t $(1 \leq t \leq 10^6)$, the number of test cases.

The next t lines contain two integers L and R $(1 \leq L \leq R \leq 10^{16})$.

Output

For each test case, print a single integer x, the answer for the interval $[L, R]$.

Example

standard input	standard output
4 1 10 10 100 100 1000 999900 999999	1 2 6 1

Note

In the first test case, the only Paper Cup that can be a Legendary Paper Cup is number 6, since $6 = 1 \times 2 \times 3$.

In the second test case, only 24 and 60 can be Legendary Copo Papels, since $24 = 2 \times 3 \times 4$ and $60 = 3 \times 4 \times 5$.

Problem M. Mayor

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

Ryei is the mayor of Chessland. His desire is to Chessland be a united and connected city. The citzens of Chessland (also known as pawns) live in their houses and they are afraid of walking through streets if it can go both ways, since they fear to be captured. For that reason, Ryei will design his streets to be one-way. He wants to connect all pawn's houses in a way that there is a directed path for every pair of houses using the fewest number of streets possible.



Ryei will construct his streets with Drex's Engineering, and they evaluated the costs of creating a street for each pair of houses. His budget is limited, so he wants to spend as little as possible. Answer the minimum total value he has to spend to achieve his goals using the fewest number of streets possible.

Input

The first line contains a single integer n ($1 \leq n \leq 10$) — the number of houses.

The i'th of the next n lines contains n integers $c_{i,1}, c_{i,2}, ..., c_{i,n}$ ($0 \leq c_{i,j} \leq 10^8$) — where $c_{i,j}$ is the cost of building a street from house i to house j.

Output

The output is a single integer C — the minimum total cost to Ryei unite his city using the fewest number of streets possible.

Examples

standard input	standard output
3 0 0 1 0 0 1 1 0 0	1
2 0 3 6 0	9