



Information

Memory limit

The limit is 512 MiB for each problem.

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most once per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Scoring

Each problem consists of several subtasks. The subtask score is awarded if all tests in the subtask are passed.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

The contestants' scoreboard is available during the contest. Use "Monitor" link in PCMS2 Web Client to access the scoreboard. The standings provided in PCMS2 Web Client are not final.



Problem A. Sheet Metal

Time limit: 2 seconds

James has a large rectangular piece of sheet metal; its width is w_1 millimeters, and its height is h_1 millimeters. This massive piece is hard to store and move around, so James and his friends want to make smaller pieces.

They found a machine that could solve this problem. The machine has a rectangular form of width w_2 and height h_2 millimeters. To use it, you put a piece of sheet metal under the form. It's allowed to have some parts of the form not be above the sheet metal. You can rotate the piece of sheet metal, but you have to make sure that the sides of both rectangles (the piece and the form) are parallel to coordinate axes.

After running the machine, everything under the form will get burnt out and be unusable. Everything that was not under the form will fall off; James and his friends must take that away with them. In addition, the remaining part might will apart into several separate pieces. If the form's edge coincides with the sheet metal's edge, we also consider that the sheet metal is disconnected there.

To ease the task of taking everything away, find a way to apply the machine to the piece of sheet metal such that **the maximum area of all remaining pieces is as small as possible**. In case there are no pieces left, print 0.

Input

The only line of the input contains four integers w_1, h_1, w_2, h_2 ($1 \leq w_1, h_1, w_2, h_2 \leq 10^6$) — widths and heights of the piece of sheet metal and the form.

Output

Print the minimum possible area of the largest remaining piece.

Scoring

Subtask	Point	Constraints
1	50	$w_2 \leq h_2 \leq w_1 \leq h_1$
2	50	No additional constraints

Examples

standard input	standard output
2 3 1 3	1.5
2 3 3 3	0
2 3 2 2	1
2 3 3 1	1.5



Problem B. Lunchtime Fruits

Time limit: 2 seconds

The school canteen has four types of fruits: apricots, bananas, apples, and pears. There are three different possible sets of fruits available for lunch:

1. two apricots, one banana, and one apple;
2. two apricots and two apples;
3. one apricot, one banana, two apples, and one pear.

The canteen's employees want to assemble as many lunch fruit sets for children as possible. Help them do it!

Since the number of fruits delivered is different every day, you need to solve the problem for multiple test cases.

Input

The first line of the input contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The following t lines contain a description of each test case. Each line contains four integers $a, b, c,$ and d ($1 \leq a, b, c, d \leq 10^9$) — the number of apricots, bananas, apples, and pears, respectively.

Output

For each test case, print the maximum number of fruit sets it's possible to make.

Scoring

Subtask	Score	Constraints
1	10	$t \leq 10; a, b, c, d \leq 10$
2	20	$t \leq 10^4; a, b, c, d \leq 10$
3	20	$t \leq 10; a, b, c, d \leq 200$
4	20	$t \leq 10; a, b, c, d \leq 10^6$
5	30	No additional constraints

Example

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

Note

In the first case, you can make two sets of fruits: one of type 1 and one of type 3.

In the second case, one of the possible answers is (2, 3).

In the third case, the optimal answer is (1, 1).

In the fourth case, you can make three sets of type 3.



Problem C. Sliding Dominoes

Time limit: 2 seconds

Let's play a game! The game field is a rectangular grid with n rows and m columns n and m are both **odd**. There are domino tiles on this grid; each domino covers two horizontally or vertically adjacent cells. At the beginning of the game, every cell, except for one, is covered by precisely one domino tile, and one cell is empty.

During each move, you can *slide* any domino tile in the direction parallel to its orientation, provided that the cell in that direction is empty. You can perform this move as many times as you like, and you can stop at any time.

Every cell on the game field has a positive or negative cost. When you slide a domino tile, some cell previously covered by that tile becomes empty. If this is the first time this cell becomes empty during the game, you add its cost to the overall score.

Figure out a sequence of moves that maximizes the total score, that is, the sum of the costs of all cells that were empty at least once during the game.

Input

The first line contains two integers n and m — the number of rows and columns of the grid ($1 \leq n, m \leq 499$; n and m are odd). The following n lines contain m characters each — the initial arrangement of dominoes. The empty cell is denoted by `.`, a horizontal tile — by a pair of characters `<` (the left cell) and `>` (the right cell), and a vertical tile — by a pair of character `^` (the top cell) and `v` (the bottom cell). It's guaranteed that the arrangement is correct and there's one empty cell.

The following n lines contain m integers each, describing the costs of cells. The cost of each cell is an integer from -1000 to 1000 inclusive. The initially empty cell has cost 0 .

Output

Print a single integer — the maximum possible final score.

Scoring

Subtask	Points	Constraints
1	10	$n = 1, m \leq 3$
2	24	$n = 1, m \leq 499$
3	14	$n, m \leq 7$
4	23	$n, m \leq 499$, all costs are non-negative
5	29	No additional constraints

Examples

standard input	standard output
<pre>1 5 <>.<> 5 2 0 9 -2</pre>	5
<pre>3 3 <>^ ^.v v<> 1 1 1 1 0 1 1 1 1</pre>	0



Problem D. Lost in Translation

Time limit: 2 seconds

This is a “run-twice” problem; your program will be run twice for each test.

Your task is to implement a program that transmits data through an unreliable channel. On the first end (during the first run), you get a binary string of length n , and your program has to be able to recover it on the other end of the channel (during the second run).

Thankfully, the channel allows you to send strings with k different types of characters ($k > 2$) and use strings of length m ($m > n$). But, there is a catch: when you transmit a string, **all occurrences of some type of character will be removed**. The remaining characters will still follow in the same order. Your job is to develop an algorithm to encode the input string, so it’s possible to recover it after the transmission.

To speed up the testing, in each test, you will need to encode and transmit t binary strings at once. These strings will be transmitted independently, and different strings might have different types of characters removed from them.

Input

During the first run, the first line contains an integer 1. The second line contains four integers t , n , m , and k — the numbers of strings to encode, the length of each string, the maximum possible length of the encoded string, and the number of types of characters you can use ($1 \leq t \leq 100$, $k = 3$ or $k = 4$).

The following t lines contain a binary string of length n .

If $k = 4$, then you can use A, B, C, D in the encoded string. If $k = 3$, then you can only use A, B, and C.

During the second run, the first line contains an integer 2. The second line contains four integers t , n , m , and k , similar to the previous description.

Then, the following t lines contain strings outputted by your program during the first run, but every string is missing some type of character. These strings are given to you in the same order as during the first run.

Output

During the first run, print t nonempty strings with at most m characters from $\{A, B, C, D\}$ or $\{A, B, C\}$, depending on the value of k .

The character to be deleted will be chosen in such a way that the resulting string is not empty. For example, if you output CCCC, all characters of type C will not be removed from the string.

During the second run, you need to decode these strings into initial binary strings of length n .

Scoring

Subtask	Points	Constaints
1	27	$t = 1, n = 10, k = 4, m = 20$
2	14	$t = 100, n = 100, k = 4, m = 200$
3	28	$t = 100, n = 100, k = 3, m = 200$
4	20	$t = 100, n = 100, k = 3, m = 190$
5	11	$t = 100, n = 100, k = 3, m = 180$



Example

standard input	standard output
1 2 10 20 4 0111011001 1111111110	BAACBBACDCDDAACCAABD DABBADCBCBBCCACA
2 2 10 20 4 AACACDCDDAACCAAD DBBDCBCBBCCC	0111011001 1111111110

Note

In the example, you need to transmit two binary strings 0111011001 and 1111111110, using strings of length $m = 20$ and $k = 4$ different characters. For example, suppose that after the first run, your program outputted BAACBBACDCDDAACCAABD for the first string and DABBADCBCBBCCACA for the second string.

Before the second run, the judges' program removes some type of character from each string. For example, here, we removed all B's from the first string and all A's from the second string. You have to restore and print the initial binary strings 0111011001 and 1111111110.



Problem E. Summer School

Time limit: 2 seconds

The Innopolis Summer School has announced its new installment. This year, the organization is a little different: there are n parallels with levels ranging from 0 to $n - 1$. Each parallel is intended to have at most k students in it.

According to their past performance and training results, each student received their approximate level — also an integer from 0 to $n - 1$.

Based on the experience of past years, the time spent studying at parallel x for a student of level L will be beneficial if the student's level differs from the parallel by at most $d + L \cdot \frac{p}{100}$. In other words, $|x - L| \leq d + L \cdot \frac{p}{100}$.

Every day the admission system is very busy: there are many new applications and cancellations. To help students plan their summer, organizers want to figure out how many students they can invite such that everyone's participation in the summer school is beneficial.

We will describe each day as one of two types of events:

1. + L $v - v$ students of level L sent their applications;
2. - L $v - v$ students of level L cancelled their previous applications.

Write a program that finds the maximum number of students the organizers can invite, so everyone's participation is beneficial for them.

Input

The first line contains integers n , k , d and p — the number of parallels, the maximum size of every parallel, and the constants that define if the study is beneficial, respectively ($1 \leq n \leq 5 \cdot 10^5$, $1 \leq k \leq 10^9$, $0 \leq d \leq n$, $0 \leq p \leq 100$).

The second line contains one integer m — the number of days you need to process ($1 \leq m \leq 5 \cdot 10^5$).

The following m lines contain descriptions of events of each day. Each line contains either + L v , or - L v , where L is the student's level, and v is the number of applications/cancellations ($0 \leq L < n$, $1 \leq v \leq 10^9$).

Output

Print m integer: the maximum number of students the organizers can invite after every change.

Scoring

Let's define C as the maximum number of applications of the same level present in the system at some moment.

Subtask	Points	Constraints
1	10	$n \leq 5, m \leq 10, C \leq 4$
2	10	$n \leq 30, m \leq 100, C \leq 30$
3	10	$n, m \leq 100, C \leq 10^6$
4	10	$n, m \leq 10^5, d = 0$
5	10	$n, m \leq 10^5, d \leq 1$
6	10	$n, m \leq 10^5, p = 0$
7	10	$n, m \leq 10^5, v = 1$
8	10	$n, m \leq 10^5$, only events of type '+'
9	10	$n, m \leq 10^5$
10	10	No additional constraints



Examples

standard input	standard output
5 2 1 25 5 + 4 7 - 4 3 + 2 5 + 3 5 - 3 2	6 4 8 8 8
5 2 1 1 6 + 0 4 + 1 3 - 0 2 + 3 7 + 4 1 - 3 6	4 6 5 10 10 7

Note

In the first example, students of level 4 can study in parallels 2, 3, and 4. That means that after the first-day event, 6 students out of 7 can study beneficially.

In the second example, every student can only study at the parallel, which level is either equal to the student's level or differs from it by at most one.