Problem A. Aaa (Division 1 Only!)

Input file:	`a.in`
Output file:	`a.out`
Time limit:	7 seconds
Memory limit:	256 Mebibytes

This and all following tasks are devoted to stories about a girl who has played an important role in the lives of the authors.

It all began like this:

During one of hundreds of absolutely unimportant lectures, senior student was evenly distributing Sudoku puzzles around herself. Sasha was also given one, and that was a preposition to speak.

— I hope this is the most difficult level?

— Just for you.

— Well, well, — seeing a deep irony in her eyes and the fact that the first three lines are already filled.

— If you solve this one, I'll give you another.

— I won't just solve this, but I will say how many solutions exist, — said Sasha boastfully, not knowing what a problem he has created for himself.

— Well, well, we'll see.

The next two days Sasha was engaged only in looking for number of solutions for sudoku. By the way, sudoku is a puzzle in which it is proposed to fill a 9×9 table with numbers from 1 to 9 so that in every row, column and each of the 9 squares 3×3, all the numbers were different. Initially, some cells are already filled and you are to write the numbers in empty cells.

— And what was the girl's name, again?

— Oh...

Input

There are 3 lines, consisting of 9 integers from 1 to 9 in each. Those are first three lines of Sudoku.

Output

The only number: number of solutions. It is guaranteed that at least one solution exists.

Example

a.in	a.out

`1 4 7 5 6 3 8 2 9`
`2 5 8 4 7 9 1 3 6`
`3 6 9 1 2 8 5 4 7`
	`7013953152`

Problem B. Bachelor pursuing (Division 1 Only!)

Input file:	`b.in`
Output file:	`b.out`
Time limit:	3 seconds
Memory limit:	256 Mebibytes

Still not knowing the name of the girl from the previous problem, Sasha decided to make inquiries, he was lucky that he had a database of university and VK at hand. It was found that Ira is an excellent student for four years and wanted to get a bachelor's degree, and as it turned out, a master's degree and PhD. And you never know what else...

— How much will it be in the sum? — Sasha thought with delight (note: in Russian for «power» and «degree» is used the same word), while subconsciously recalling a recent problem from the contest which he has not solved. Two numbers are given: N and K, compute

$$\sum_{i=1}^{N} i^K$$

Input

The first line contains Q, the number of queries. Each of the following Q lines contains two numbers N and K.

$(1 \le Q \le 41\,000,\ 1 \le N \le 10^9,\ 1 \le K \le 1000)$

Output

For each query on a separate line write one number — the sum of K-th powers of positive integers from 1 to N modulo $10^9 + 7$.

Example

b.in	b.out
4 2 1 4 2 3 3 11 11	3 30 36 675987247

Problem C. Concatenation of credits

Input file:	`c.in`
Output file:	`c.out`
Time limit:	1 second
Memory limit:	256 Mebibytes

For her outstanding achievements, Ira was entrusted to read a lecture to students of lower courses. Having good experience in training on lectures of extraneous stuff (like ones from problem A), Ira found another thing to do. Using the fact that students were writing a test and gave her their student's record-books, she began to look for patterns in their marks.

I must say that teachers at the university are tough. First, they'll never give you a 100 mark. And secondly, they never give you a mark which already exists in your student's record-book.

Thus, in each student's record-book, Ira saw six different ratings from 10 to 99. And instead of putting there the seventh, she concatenated the six other marks and divided them by her favorite number. Sasha, who was watching the scene, said:

— You know, there are pretty many combinations of marks evenly divisible by your favourite number.

— Hm... About three? — Ira said with ill-concealed derision.

— No. I don't think we have enough students in university.

— Well, well.

Input

Only one number is given: Irina's favorite number I ($1 \le I \le 100$).

Output

Display the number of ways to choose an ordered six different two-digit numbers such that their concatenation is evenly divisible by I.

Example

c.in	c.out
10	44828253360

Problem D. DeviantArt (Division 1 Only!)

Input file:	`d.in`
Output file:	`d.out`
Time limit:	4 seconds
Memory limit:	256 Mebibytes

The art of photography is another Ira's hobby. Her works occupy a worthy place on the pages of DeviantArt. Every Internet user can freely go to her gallery and «Like» image(s) he liked, increasing rating of those image(s) by 1.

— Mmm! Not a bad job for a bot.

The essence of the bot consisted in the following: a specially created user «Likes» K photos, starting with I-th, in increments of A. The second bot was created to monitor results. It checks the sum of ratings for L pictures, starting with J-th, in increments of B. Photographs are numbered from 0.

Bots work independently until the administration of DeviantArt will notice the cheating with «Likes» and bans both Anonymouses.

Input

First line of input file contains four integers N, A, B and Q — number of pictures, size of steps for bots that do not change throughout their lives, and the total number of iterations, respectively. Then in each of Q of lines written a command for a bot:

`s` I K — first bot «Likes» K pictures, starting from I-th.

`g` J L — second bot counts rating of L pictures, starting from J-th.

Before bots started their actions, all pictures had rating 0.

($1 \le N \le 10^5$, $1 \le A \le N$, $1 \le B \le N$, $0 \le Q \le 10^5$;

$0 \le I \le N-1$, $1 \le K \le N$, $I + (K-1)A \le N-1$;

$0 \le J \le N-1$, $1 \le L \le N$, $J + (L-1)B \le N-1$)

Output

In a separate line print a result for each command for second bot.

Example

d.in	d.out
10 2 5 7 s 0 5 g 0 2 g 1 2 g 2 2 s 1 3 g 3 1 g 4 2	1 1 1 1 1

Problem E. Exam

Input file:	`e.in`
Output file:	`e.out`
Time limit:	1 second
Memory limit:	256 Mebibytes

Dislike of philosophy brings people together, the need to take the exam especially. But the main thing is a clear sense of reluctance to prepare for it.

— See, Ira, you come to the teacher and propose him such a thing: we have 18 tickets, they lie face down. Let us turn over K of them, then you have them all properly shuffled and put in one pile. After that I, I mean you, close your eyes and have to divide them into two piles so that they will have the same number of tickets with their faces up. Then you are allowed to swap any two tickets in the first pile, turn over one ticket in the first pile, and permanently move the tickets from the first pile to the second. You're doing it with your eyes closed and don't have any information about the status of tickets other than that you know the number K and remember all your actions.

— Okay, so what?

— So, if you will succeed, the teacher gives you the opportunity to answer any of the open tickets.

— And if not?

— We'll have to believe in a miracle.

— Why don't you try it yourself?

— Not that I have a good reputation.

— So you want to spoil mine?

— Oh, three times.

Input

Given only an even integer K ($0 \le K \le 18$) — the number of cards flipped face-up (initially all 18 tickets are in the first pile).

Output

If you can not separate tickets into two non-empty piles as needed, your output should be -1.

Otherwise, the first line should contain the number of actions Q to make ($0 \le Q \le 2^9 + 36$). Further, in Q rows, describe the procedure. Each line should contain the command of one of three types:

`swap i j` — swap tickets at positions i and j in first pile.

`rev i` — turn over the ticket in position i in first pile.

`out i` — Move a ticket in position i from the first pile to the second. After this operation, positions all tickets in the first pile, starting from the $(i + 1)$-th, will decrease by 1.

Piles should be non-empty and contain the same number of open tickets, $1 \le i \ne j \le 18$, and both i and j should be less than current size of first pile.

e.in	e.out
18	9 out 1 out 1 out 1 out 1 out 1 out 1 out 1 out 1 out 1

Problem F. Fate to hate

Input file:	`f.in`
Output file:	`f.out`
Time limit:	1 second
Memory limit:	256 Mebibytes

Ira's birthday was approaching. On the night from 27th to the 28th, Sasha could not sleep for a long time. Mathematically logical nonsense was discretely coming to the head. An ordinary daisy divination has become a cruel game of numbers and bit operations.

The dream was about an infinite field of indistinguishable daisies. Each daisy had N petals, and each of them had a number on it. From time to time, the fear that Ira can change the number of her ICQ forced to wake up in a cold sweat. And every time a new number came to head, it certainly had to be obtained from the petals by applying to the numbers written on them operations, AND and OR. If the number could be obtained, it was a good sign, otherwise the same dream became infused with horror of hate and evil.

All daisies are the same, and they can be endlessly to picked in order to use the right petals. So to get one number, you can pick several daisies and pick a non-empty subset of their petals. In his dreams, Sasha is able to fully control the order of operations.

Input

The first line contains the number N — the number of petals on a daisy. The next line lists the integers on the petals a_i. Next line contains the number Q — the number of ICQ-numbers which must be obtained. Each of the following Q lines contains one ICQ number b_j.

($1 \le N \le 10^5$, $0 \le a_i \le 10^9$, $1 \le Q \le 10^5$, $0 \le b_j \le 10^9$).

Output

For each ICQ number, output one line containing "`Fate`" if this number can be obtained from the numbers on the petals by applying bitwise AND and OR operations to them, or "`Hate`" if it is impossible.

Example

f.in	f.out
3 1 4 5 6 1 2 3 4 5 6	Fate Hate Hate Fate Fate Hate

Problem G. Genealogic tree (Division 1 Only!)

Input file:	`g.in`
Output file:	`g.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

There was a lot of relatives on Ira's birthday party, so that to remember them all was not so easy task. In order to obtain an overall picture, the idea to make a family tree was born. Due to error in one of tree building procedures it turned out that the tree was not binary, although the number of vertices was correct: $2^k - 1$.

— Something is wrong here.

— Really?

— I propose to double-check everything.

— Specifically.

— Divide the tree into connected parts of 2^i vertices, and each of us will check his part.

— Why do you want to do that this way?

— I do not know, I just like powers of two, and also the sum will be exactly what we need. Although there is a little problem: you can do that in a bunch of ways.

— Yes, I remember the student's record-books, and Sudoku. How many ways are here this time?

— You have a great memory, give me a few minutes to think about it.

— Catch it.

Input

The first line contains an integer $N = 2^k - 1$ — number of vertices in the tree. The next line contains $N - 1$ integers. Integer a_i on i-th position denotes that there is an edge between vertices $i + 1$ and a_i. Vertices are numbered starting from 1.

$(0 \le k \le 12,\ 1 \le a_i < i + 1)$

Output

Print a single integer — the number of ways to divide the tree into exactly k connected blocks of sizes 1, 2, 4, ..., 2^{k-1}. Each vertex must belong to exactly one block.

Example

g.in	g.out
7 1 1 2 2 3 3	4

Problem H. Happy Birthday (Division 1 Only!)

Input file:	`h.in`
Output file:	`h.out`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Then came the long-awaited birthday. Only one test left — find a way to Ira's home.

The most popular form of transport in the city are buses. Each bus has two drivers, one of them knows one route, and the other — other route. However the time and place of start are the same for both bus drivers. Each day one of the drivers is working, and the other is resting. If Alex is on a bus stop, he will immediately know which drivers are now working on buses departing from this stop. However, before he reaches the stop, he only knows the schedule of possible movement of buses and the probabllity that the first driver works today. Sasha lives near a bus stop with number 1 and can be on it at any time, and Ira lives near to the stop N. Find the average arrival time for Sasha. At the same time Sasha would never use a way that could cause not reaching Ira's home at all, but among all the other he chooses the one that minimizes the expected time of arrival.

Important facts:

Network traffic of buses is an acyclic directed graph.

You can instantly change from one bus to another one.

In determining the optimal strategy Sasha also counts that on arrival at each stop, he gets the informations about today's drivers.

Initially, Sasha doesn't know extra information even about his stop.

Input

In the first row are given two integers N and K — the number of stops and buses. In each of the next K lines information about buses is described: seven integers u, d, p, v_1, a_1, v_2, a_2 — number of stops and time of departure, the probability that the first driver works, place and time of arrival, for first and for second driver, respectively.

($2 \le N \le 10^5$, $0 \le K \le 10^5$, $1 \le u, v_1, v_2 \le N$, $u \ne v_1$, $u \ne v_2$, $0 \le d, a_1, a_2 \le 1440$, $d < a_1$, $d < a_2$, $0 < p < 100$)

Output

Print average arrival time to N'th stop for Sasha with an absolute or a relative error of 10^{-6}, or -1, if there is a nonzero probability of not getting there at all.

Example

h.in	h.out
5 6 1 60 50 2 200 3 150 1 100 25 2 160 3 150 1 200 50 5 350 4 300 2 180 50 5 300 4 280 3 400 80 5 600 5 660 4 350 50 5 500 5 550	423.4375000000000

Problem I. I love Ira (Division 1 Only!)

Input file:	`i.in`
Output file:	`i.out`
Time limit:	12 seconds
Memory limit:	256 Mebibytes

Sasha wanted to make Ira a very unusual birthday gift. Romantic message posted with solved problems, on the most popular Ukrainian online judge. Problems on the site are in the form of a rectangular table, solved problems are highlighted in a pleasant green. This has suggested the idea to light in the table only those cells that would form a desired image or inscription...

Even if the shape of the message is strictly defined, its position in the table can be controlled. Of course, on the location of the signs depends set of tasks to be solved. So Sasha has assigned to each task a score — a integer from 0 to 9, and decided to location in such way, so that the sum of marks would be maximial. Text can be placed in a table anywhere, with only one limitation: inscription must entirely lie in the table.

Input

First line of the input file contain two integers R and C — number of rows and columns in the table, respectively. In next R lines map of the problems is described. Each line has exactly C digits — score for corresponding tasks. The score for the task is measured by the integer from 0 to 9. The next line gives two numbers H and W — height and width of the inscription, respectively. In next H lines template of the inscription is given. Each line consists of W characters: '`#`' means that the problem needs to be solved, '`.`' means that the problem should not be solved.

$(1 \le R, C \le 800,\ 1 \le H \le R,\ 1 \le W \le C)$

Output

Print the maximum sum of scores for solved problems.

Example

i.in	i.out
22 18 000000000001000000 003001000000000000 000000000000000003 020000000010000000 090000000901000100 300400001911323500 000000000000000000 000000000000000000 000001000001000049 020100113210100000 200000006220004100 600009000217100009 000001000031004210 020101600100000190 110001002000000000 000092096071021990 000094000120010100 201003007430040300 100000010000100000 101040008000020001 404000000040043200 000000010000001000 17 18 ###...#........... .#....#........... .#....#..#..#.#.## .#....#.#.#.#.#.#. .#....#.#.#.#.#.## .#....#.#.#..#..#. ###...#..#...#..## ....###............. ...#.............. ...#..#.##.##..... ...#...#.#...#.... ...#...#....##.... ...#...#...#.#.... ..###..#...##.#...	69

Problem J. Justice

Input file:	`j.in`
Output file:	`j.out`
Time limit:	1 second
Memory limit:	256 Mebibytes

— Let's play a game. There are several heaps, each of them contains certain number of stones. In one move you can take any number of stonesfrom one of the heaps. If you cannot make a move — you lose.

— Well OK, but I go first.

— Okay, then I choose how much stones we have.

— Okay, then I choose how much heaps we have.

— Then I will divide stones into heaps.

— Good luck.

Input

Two numbers N and K — the number of stones and the number of heaps.

($1 \le N \le 10^9$, $2 \le K \le 16$)

Output

If you can not divide the N stones smoothly on K non-empty heaps, so that the optimal game of both players leads to win of a second player, print -1. Otherwise, print K positive integers a_i — the size of heaps.

Example

j.in	j.out
4 2	2 2

Problem K. Knife to me (Division 2 Only!)

Input file:	`k.in`
Output file:	`k.out`
Time limit:	1 second
Memory limit:	256 Mebibytes

Birthday cake was very good, tender coffee cake with layers of cream, filled with white chocolate — a miracle of culinary art. But there is a problem: cake is shaped like a prism whose base is the regular N-angled polygon — just as the number of guests, and arrival of Sasha complicated the situation. Now, the cake should be divided into $N + 1$ part.

— Who of us likes geometry?

— Well, you studied at math faculty, didn't you?.

— Amazing arrogance. You're to blame — and you are gonna take the knife.

— Then, I propose to make vertical cuts.

— Well, you're right, Captain Obvious.

— It's not all. Divide evenly on the $N + 1$ parts so that the volumes of all pieces of the same.

— Well, ok.

— It's still not all. Divide so that the surface area, filled with chocolate,on all the pieces, was the same.

— It was easier to cut on my own.

We consider the projection of the cake on the table. It turns a regular polygon centered at $(0, 0)$ with one vertex at the point $(1, 0)$. The required pieces must be convex polygons such that no three vertices of a piece lie on a straight line. The plan must satisfy the cutting conditions, regardless of the height of the cake. Recall that the chocolate is only on the outer surface of the cake, meaning the upper and side faces.

Input

The number N — the number of guests except for Sasha ($3 \le N \le 100$).

Output

Print exactly $N + 1$ block, each of which describes a single piece. The description of a piece begins with K_i — number of vertices ($3 \le K_i \le 100$). In the next K_i lines, print vertices in the anti-clockwise order, one vertex per line, with possible inaccuracy not greater than 10^{-6}.

Example

k.in	k.out
3	3 0.0000000000000 0.0000000000000 1.0000000000000 0.0000000000000 -0.1250000000000 0.6495190528383 4 0.0000000000000 0.0000000000000 -0.1250000000000 0.6495190528383 -0.5000000000000 0.8660254037844 -0.5000000000000 0.0000000000000 4 0.0000000000000 0.0000000000000 -0.5000000000000 0.0000000000000 -0.5000000000000 -0.8660254037844 -0.1250000000000 -0.6495190528383 3 0.0000000000000 0.0000000000000 -0.1250000000000 -0.6495190528383 1.0000000000000 -0.0000000000000

Problem L. Lie to me (Division 2 Only!)

Input file:	`l.in`
Output file:	`l.out`
Time limit:	1 second
Memory limit:	256 Mebibytes

— Nothing personal, but I don't believe in your ways of cutting the cake.

— You'd better believed.

— Than prove it.

— I'm not so cool, but my notebook never lies to me.

— I can't see a backpack on you.

— Damn it, haven't I taken it?.. Joke, surely I have.

Reminding that the cake cutting plan contains the following information: The cake is represented as regular N-gon with center at point $(0, 0)$ and one of the vertices at point $(1, 0)$. Whole cake is cut into $N + 1$ pieces. Each of those pieces is convex polygons.

Check if all the pieces are convex polygons without three vertices on one line, in counterclockwise pass order. Check if all the pieces have equal square. Check if all the pieces have equal length of the outer border, i.e. the part of the border that reproduces the initial cake border. Check if is possible to join all the pieces together, so they will form the full cake.

Input

In the first line is given number N — number of the cake vertices. Then the description of $N + 1$ pieces is given. The description of one piece is started with number K_i — number of the vertices in the piece. Next K_i lines contain those vertices listed in random order, given by two coordinates x_i and y_i ($-100 \le x_i, y_i \le 100$, $3 \le N \le 100$, $3 \le K_i \le 100$). All the coordiantes are real numbers with not more than 15 decimal places.

Output

Print "`Yes`" in case the description matches the correct cake cutting, otherwise "`No`". It's guaranteed that in case the answer is "`No`" parameters which provide this answer are not less than 10^{-6}.

Example

l.in	l.out
3 3 0.000000000000 0.000000000000 1.000000000000 0.000000000000 -0.125000000000 0.649519052838 4 0.000000000000 0.000000000000 -0.125000000000 0.649519052838 -0.500000000000 0.866025403784 -0.500000000000 0.000000000000 4 0.000000000000 0.000000000000 -0.500000000000 0.000000000000 -0.500000000000 -0.866025403784 -0.125000000000 -0.649519052838 3 0.000000000000 0.000000000000 -0.125000000000 -0.649519052838 1.000000000000 -0.000000000000	Yes

Problem M. MySpace (Division 2 Only!)

Input file:	m.in
Output file:	m.out
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Situation similar to the one in the problem D occured on MySpace website.

Briefly, two bots had to increase photos rating — Each «Like» on the picture increases its rating by 1. First bot «Likes» all the photos with numbers K such that $K \bmod A = I$. The second bot counts the rating of all the pictures with numbers L such that $L \bmod B = J$. Pictures are enumerated starting from 0.

Input

Four numbers N, A, B and Q are given in the first line — number of the photos, bot step values, which don't change during their life, and total number of iterations. In every next Q lines command for the bot is written:

`s` I — first bot «Likes» photos $I, I + A, I + 2A, I + 3A \ldots$

`g` J — second bot sums «Likes» of photos $J, J + B, J + 2B, J + 3B \ldots$

Before bots start their activity, all photos had rating 0.

($1 \le N \le 10^5$, $1 \le A \le N$, $1 \le B \le N$, $0 \le Q \le 10^5$, $0 \le I < A$, $0 \le J < B$)

Output

For every command of the second bot, print its result on a separate line.

Example

m.in	m.out
10 2 5 7 s 0 g 0 g 1 g 2 s 1 g 3 g 4	1 1 1 2 2

Problem N. Now (Division 2 Only!)

Input file:	`n.in`
Output file:	`n.out`
Time limit:	1 second
Memory limit:	256 Mebibytes

In every sense, it's winter now. Only memories of the past could make us see the clear view of things.

— Is it really the end?

— Who knows...

— Let's hurry up!

The undirected graph without loops and multiple edges is given. Find the size of the maximal matching, i. e. maximum size of the subset P of the graph edges, considering that every vertex has no more than one edge from P.

Input

In the first line you are given two integers N and K — number of verticles and edges in the graph. Every line of the next K contains two integers u and v — description of one edge. Graph is guaranteed to be rather random.

$(1 \le N \le 400,\ 0 \le K \le \frac{N(N-1)}{2})$

Output

Print one integer — the size of the maximal matching.

Example

n.in	n.out
5 5 1 2 1 3 1 4 1 5 2 3	2