Problem A. You're in the Army Now

Input file:	army.in
Output file:	army.out
Time limit:	3 seconds
Memory limit:	256 megabytes

Orcs are proud of their army. And all orcs, no matter, men or women, want to join the army. But to join it, you need to pass strong exams. An orc should pass running, jumping, swimming, shooting and so on. For every of these exams, the orc gets some points. The more points the orc gets, the better chance he has to be join the combat arm he wants. To satisfy every orc, the warchief declared new system of joining the army. As we know, there are M combat arms. Every orc writes up to K combat arms he wants to join, in the order from the combat arm he prefers the most to the one he prefers the least. Then, the following rules are applied:

1. If some orc joins some combat arm, there is no combat arm with higher priority for him he can join. This means that for every combat arm with higher priority, the least total number of points gained by orc in that combat arm is greater than his points.
2. If some orc is not put into any combat arm, rule 1 is correct for every combat arm he applied to.
3. For i-th combat arm at most a_i orcs are put into it.

So now the orc warchief wants to know which combat arm every orc will join. Help him to do that.

Input

In the first line of input file there are three numbers: N, M and K — number of orcs, combat arms and maximal number of priorities for every orc ($1 \le N \le 10^5, 1 \le M \le 10^5, 1 \le K \le 10^5$). On the second line there are M numbers a_i — maximal number of orcs that can be put into that combat arm. These numbers are positive and do not exceed N. Latter on N lines there are definitions of orcs. Every definition consists of orc's name, sum of his points, number of his priorities and priorities themselves (from highest to lowest). Orc name can consist only from lowercase and uppercase Latin letters and spaces, but does not have leading and trailing spaces (you should omit them if they're in the input). Length of every name does not exceed 20 characters. Sum of orc's points is positive integer number not exceeding 10^9. Number of priorities do not exceed K and sum of these numbers do not exceed $2 \cdot 10^5$ for all input file. Priorities are numbers between 1 and M. You can assume that there are no two orcs with the same sum of points.

Output

For every combat arm you should output list of orcs which will be put into it in alphabetical ordering (space is considered less than any letter). When letters are equal consider capital letters less, in other case don't consider their case. Output lists in order from combat arm numbered 1 to combat arm numbered M. Separate lists by blank line.

Example

army.in	army.out
4 2 2 2 2 Grom Hellscream 1200 2 1 2 Thrall 1500 2 1 2 Orgrim Doomhammer 1450 2 2 1 Kargath Bladefist 1400 2 1 2	Kargath Bladefist Thrall Grom Hellscream Orgrim Doomhammer

Problem B. Art

Input file:	art.in
Output file:	art.out
Time limit:	2 seconds
Memory limit:	256 megabytes

Orcs also like art. Recently Thrall the warchief abducted some goods from caravan along with great Kalevich's artwork "Beauty and the Beast". There is legendary shaman Ner'zhul with some ugly human beings on it. Thrall the warchief respects those great shaman, so he wants to tack it to the wall at home. But humans are realy disgusting! So Thrall asked you to crop this picture in order to remove all ugly parts of it. Of course, it is forbidden to crop any part of legendary shaman. Also, original aspect ratio must be kept and cropped picture must be as large as possible. So you have to choose some rectangular part of the original image with the same aspect ratio in such way what image of Shaman is fully represented on it and no part of ugly humans are.

Input

First line of input file contains 2 integer numbers W, H — size of famous Kalevich's artwork ($1 \le W, H \le 1000000$). In the second line there are four integer numbers x_1, y_1, x_2, y_2 — coordinates of bounding box's bottom-left and top-right corners for image of great shaman Ner'zhul ($0 \le x_1 \le x2 \le W$ and $0 \le y_1 \le y_2 \le H$). Then integer number N is present ($N \le 10000$). In the following N lines will be given 4 integer numbers in each: x_1, y_1, x_2, y_2 — coordinates of bounding box's bottom-left and top-right corners for ugly images. In this task coordinates of bottom-left corner of artwork are $(0, 0)$ and coordinates of top-right corner are (W, H). You may assume, that image of shaman occupy excacly all bounding box. The same is true for human beings on picture.

Output

In the first line of output file write out single real number — maximal possible scaling coefficient. In the second line write out two real numbers — coordinates of bottom-left corner of cropping rectangle. If it is impossible to fullfil wish of the warchief, write out "-1" in the only line of output file. Your answer must be accurate up to 10^{-6}.

Examples

art.in	art.out
6 10 2 4 4 7 4 0 0 1 2 4 0 6 2 4 7 6 10 0 7 1 10	0.5 2.0 1.0
10 10 0 0 5 5 2 4 4 5 5 2 6 10 7	-1

Problem C. Ballistic

Input file:	`ballistic.in`
Output file:	`ballistic.out`
Time limit:	1 second
Memory limit:	256 megabytes

In the army of Orcs in order to calculate trace of ballistic shell they have to solve complicated systems of first order linear ordinary differential equations with constant coefficients. They know the theory well, but they lack processing power to do it (you know, abacus is not suitable to do such calculations). So they ask you to help them find characteristic polynomial of coefficient matrix. This polynomial is determined by following formula: $|A - \lambda \cdot E|$, where A is given matrix, E is identity matrix and λ is variable.

Input

In the first line of input file there is one integer number N — size of coefficient matrix of system of first order linear ordinary differential equations ($1 \le N \le 20$). In the following N lines there will be given N integer numbers — coefficients themselves. All numbers do not exceed 10^3 by absolute value.

Output

In $N+1$ lines of the output file write out $N+1$ integer numbers — coefficients of characteristic polynomial from highest power to lowest.

Examples

ballistic.in	ballistic.out
3 1 2 3 4 3 6 7 8 8	-1 12 42 29
2 1 2 2 1	1 -2 -3

Problem D. Castle

Input file:	`castle.in`
Output file:	`castle.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

In the past there was big and mighty kingdom called Lordaeron. It was ruled by wise king Terenas. Lordaeron had mighty army and a lot of castles and fortresses. To control such army Terenas appointed several generals from people, which he trusted. One day first general said: "My king, this is unsafe to leave our main castle without good defense. Let me build a wall around it.". Of course, the king agreed (well, one more wall could not make things worse). After the wall was completed, other general said: "My king, this is still unsafe to leave our main castle without good defense. Let me build another wall around it." The king agreed. After the second wall was completed one other general said the same words... After all, N walls were built around the castle and all area between walls was filled with fields. Once upon a time enemies of Lordaeron had sieged the main castle. What they could do with walls? They couldn't climb on them. But they had bows and arrows. They knew also that everything between walls would burn well. During the siege, intelligence of Lordaeron discovered all aimed points of shoots. And king Terenas was able to count area, burned by enemies. Could you do it nowadays?

You may consider following facts. The castle is very small and may be considered as point in origin. All walls are convex polygons, they do not touch each other or intersect, and origin point is always inside all polygons. Aimed points are not on the walls. If aimed point is strictly outside the outer wall, then nothing will burn. Otherwise, all area that can be reached form aimed point without crossing walls will burn. Your task is to calculate area that was burned.

Input

In the first line there is number of walls N ($1 \le N \le 10^5$). Then N blocks follow. Each block contains first line with one number K — number of vertices in polygon which represents the wall. Next K lines follows each contain two integers X and Y — coordinates of polygon vertice. Each polygon will be given in counterclockwise order. Each polygon will be non-degenerate (will have nonzero area). You may assume, that total number of points in all polygons will be not greater than 10^5. Walls are given in random order.

After that follows number of aimed points, M ($0 \le M \le 10^5$). Then M lines contain coordinates of M aimed points.

All coordinates in input file are integers and are not greater than 10^6 by absolute value.

Output

Print only one number with at least five digits after decimal point — total area that was burned.

Example

castle.in	castle.out
3 4 -10 -10 10 -10 10 10 -10 10 4 20 20 -20 20 -20 -20 20 -20 4 30 -30 30 30 -30 30 -30 -30 3 1 1 22 23 111 123	2400.000000

Problem E. Creeping

Input file:	`creeping.in`
Output file:	`creeping.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

The legendary paladin Lord Uther the Lightbringer is in trouble. During head injury he lost all his experience. So he need to restore his past power. The only way to get experience is to kill creeps (usually hostile monsters). There are N creeps in this world living in different places. Each creep is characterised by his initially hit points (h_i), damage per second it can deal to Uther (d_i) and experience gained from killing it (e_i). Uther can kill creeps in any order. He have unlimited time to restore his experience, so he does not consider total time of creeping. Uther has K levels. Each level is characterised by Uther's max hitpoints (H_j), damage per second dealt by paladin (D_j), hitpoint restore rate per second (R_i) and experience threshold (E_j). So then Uther's experience reaches E_j, he gains level-up, his hitpoints value changes by difference of H_j and H_{j-1} (if he had full hitpoints he will also have full hitpoints after level-up). Also his max hitpoints, damage per second, and hitpoint regeneration rate will change to new ones. The fights are done in the following way: each second Uther and creep deal damage to each other and Uther's hitpoints are regenerated by corresponding number. If somebody has non-positive number of hitpoints, he dies. For example, if Uther has regeneration rate 4, creep can deal damage of 3 hitpoints per second, Uther will never die. After killing a creep Uther gains corresponding number of experience, and can get a level-up. If he gets several level-ups simultaneously, his characteristics will change to last ones at once. If Uther engages somebody, he cannot run away. You must note, that maximal hitpoints on the next level could be less than on the current level. Your task is to help him determine maximal possible experience he can gain and in wich order he have to kill creeps to do it. Of course, Uther does not want to die. In the beginning, Uther have zero experience and first level.

Input

First line of input file cantains two integer values N and K — number of creeps, and number of levels ($1 \le N \le 20$, $1 \le K \le 100$). Next K lines contain four integer values in each: E_j, H_j, D_j and R_j — characteristic of corresponding level ($1 \le j \le K$). Next N lines contain descrition of creeps — three integer values in each: h_i, d_i and e_i ($1 \le i \le N$). All characteristic are positive, less than 10^6 and $0 = E_1 < E_2 < \cdots < E_K$.

Output

It the first line of output file write out maximal possible experience, Uther can gain. In the second line print p — number of creeps Uther must kill to gain such experience. And in the last line write out p integers — numbers of creeps in order Uther are to kill them (creeps are numbered from 1 in order they are given in the input file).

Example

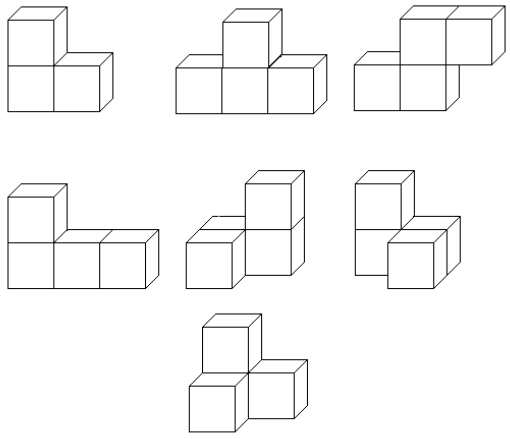
creeping.in	creeping.out
3 5 0 10 2 1 10 20 3 2 20 30 5 2 30 40 7 3 60 55 10 4 10 3 10 10 2 10 25 5 100	120 3 2 1 3

Problem F. Lock

Input file:	`lock.in`
Output file:	`lock.out`
Time limit:	3 seconds
Memory limit:	256 megabytes

Tyrande Whisperwind wants to free Illidan Stormrage from his imprisonment. She thinks he can help night elves to fight the burning legion. Not all elves agreed with such her decision, but she prefers to act instead of talking. After several fights with dungeon denizens she and her warriors at last came to gate which leads to cage room. The gate is very big and seems to be sealed by magic so it is impossible to open it by brute force. But Tyrande does not give up. She found that to open this gate she needs a key, which was disassembled right after sealing the gate. Of course, Tyrande managed to find all parts of the key, but now she's in trouble: how to assemble the key back?

The key can be imagined as a 3-dimensional figure, consisting of unit cubes, connected via their faces. So, the parts of the key can be imagined in the same way. There are 7 parts of key. One of them has 3 cubes, and six other consinst of 4 cubes. All of these parts are different (that means that none of them can be made by rotating of some other one). For more details take a look at the sample picture below.



Descritpion of 7 parts of the key.

The key itself is described as figure consisting from unit cubes. This figure will be described in M blocks. Each block will have N lines, each line will have K numbers in it. k-th number in j-th line of i-th block is 1 if cube with coordinates (i, j, k) presents in the key, and is 0 otherwise.

Input

You will be given three integer numbers M, N and K in first line. $(0 < N, M, K < 20)$. Then follow M blocks each consisting of N lines with K integers 0 or 1 in each.

Output

Output any possible assembling of the key. Output M blocks, each with N lines, K characters on each line. k-th character in j-th line of i-th block must be '0', if this unit cube does not belong to any part, or number of the part of the key this unit cube belongs to. You may number the parts in any order with numbers $1 \ldots 7$. If there are more than one solution, output any one. Do not consider process of assembling (this is done by magic, so it is not necessary that your assembling could be done in real world). Blocks must be separated with single blank line.

Example

lock.in	lock.out
3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	112 312 444 562 367 347 552 566 377

Problem G. Princess

Input file:	`princess.in`
Output file:	`princess.out`
Time limit:	1 second
Memory limit:	256 megabytes

Evil witch imprisoned the princess in the big-big tower. And once, handsome prince came to rescue the princess.

— Princess, I came here to rescue you, — he said.

— I'm so happy, my prince. But there is only one window in this horrible tower. How will you put me down? — Princess asked.

— Oh... Let's use your hairs as a rope.

— Good idea, — princess said. She cut her hair and lower them from window. But the hair were not long enough!!!

— Prince, let's wait until my hair is long enough! — princess said.

— So, will I hung around for some time, Ok? — prince asked.

— Uhmm...wait, I will tell you now when will hair be long enough.

And princess asked you by a magic mirror, how long it will take to get hair with length L? You may assume that at this moment princess has hairs with length 0 meter and they grow up with speed 1 meter/week. And princess can do two things:

1. humbly grow up her hair
2. cut them

If she cuts her hair when the length of them is K meters, hair will grow up with speed K meters/week since this moment. You must assume that princess doesn't know how to concatenate hair, so she need to grow up hair with length not less than L.

Input

One integer number L ($1 \le L \le 10^{18}$).

Output

One real number with six digits after decimal point — minimal number of weeks that it will take princess to grow up her hair to length L.

Example

princess.in	princess.out
1	1.000000

Problem H. Runes

Input file:	`runes.in`
Output file:	`runes.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Jaina Proudmoore is a very strong human mage. She could not became so strong without everyday trainings and a lot of studying. Now she is studying runes. Recently she has found a very interesting cave with lots of runes on its walls. Every rune there represents a set of circles drawn on a wall. Jaina thinks that the meaning of a rune can depend on some characteristics of picture, obtained by drawing this rune. Now she wants to investigate this. After some research she decided to figure out areas of finite parts, which appear after drawing rune. She does not need to know the parts themselves, only their areas. For example, if rune consists of two intersecting circles, there will be three finite parts.

Input

The first line of the input file contains single number N ($1 \le N \le 50$) — number of circles forming the rune. Then N lines with circles' descriptions follow. Each description consists of three numbers: coordinates of circle center and its radius. All numbers are integers not exceeding 1000 by absolute value. Radius is always positive.

Output

At the first line output P — the number of finite parts. On the second line output areas of finite parts in increasing order. Your answer must be accurate up to four digits after decimal point. Do not output parts, which areas are less than 10^{-4}.

Example

runes.in	runes.out
2 0 0 2 2 0 2	3 4.913479 7.652892 7.652892

Problem I. Spell

Input file:	`spell.in`
Output file:	`spell.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Troll witch doctor Vol'jin is finding better spells for his healing magic. Now he is trying to combine ancient spells with modern knowledge to make his spells heal faster than ever known. Vol'jin took N spells from ancient books. He wants every his new spell to be part of all of these ancient spells. This means, that if you will pronounce the ancient spell, Vol'jin's new spell will be also pronounced. Witch doctor codes spells with small latin letters to make them easy to write. Of course, there can be a lot of different spells, which are part of every ancient spell, chosen by Vol'jin. So he doesn't want you to find all of them, instead of this he wants to get such spell by it's number in lexicographical ordering of all such spells. Your task is to find spells wanted by Vol'jin.

Input

On the first line of input file there is only number N ($1 \le N \le 20$) — number of ancient spells chosen by Vol'jin. On the following N lines there are descriptions of these spells. Each description is a string of small latin letters. Sum of lengths of all descriptions do not exceed 10^5. After that the line containing M — number of spells Vol'jin wants to check follows. Then follow M lines with a single number in each — one-based position in lexicographical ordering of spell Vol'jin wants to check. It is guaranteed that this position does not exceed total number of suitable spells.

Output

For every Vol'jin's question output corresponding spell. Output every spell on single line. It is guaranteed that the total length of all answers will not exceed 10^5.

Example

spell.in	spell.out
2 abacaba dabacaba 16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	a ab aba abac abaca abacab abacaba ac aca acab acaba b ba bac baca bacab

Problem J. Orcish Transportation

Input file:	`transportation.in`
Output file:	`transportation.out`
Time limit:	1 second
Memory limit:	256 megabytes

As we know, the orcs came to Azeroth through the Dark Portal. And now they're living in their new home in Kalimdor, called Durotar. They live in peace with humans and have a lot of business and cultural contacts with them.

Both orcs and humans have built a lot of towns to live in. And both of them have built many roads to trade. It can seem strange, but the roads are unidirectional. Every road can be characterized by its capacity — the amount of goods that can be transported along it in a unit of time.

By a strange coincidence, the number of human towns is equal to the number of orc towns. By one more strange coincidence, we can construct a correspondence between human and orc towns with the following properties:

- For every orc town, there is one and only one human town corresponding to it.
- The orc capital corresponds to the human capital.
- If there exists a road from town A to town B there always exists a road from town B' to town A' with the same capacity (here, T' denotes a town corresponding to town T).

So, we shall extend this correspondence between towns to a correspondence between roads.

The trade between orcs and humans is growing more and more profitable. Recently, the orc leader thought about the amount of goods that can be transported in a unit of time between the capitals of orcs and humans. It would not be a problem for him, but there are some restrictions for trading. First of all, the transportation on every road can be done only in one direction (because roads are unidirectional). Second, the amount of goods transported along every road in a unit of time should be equal to the amount of goods transported along the corresponding road, and, of course, this amounts should not exceed the capacities of the roads. And the last restriction is that for every town except the capitals, the amount of goods transported to that town in a unit of time should be equal to amount of goods transported from it.

Now, the orcs leader wants to know what is the maximal amount of goods that can be transported between capitals under all these restrictions.

Input

On the first line of input there are two integers N and M separated by a space — the number of orc (and human) towns and the number of roads, respectively ($1 \le N \le 300$, $1 \le M \le 50\,000$). Orc towns are numbered from 1 to N and human towns are numbered from $N + 1$ to $2 \cdot N$; the human town that corresponds to the orc town number i is numbered $i + N$. The orc capital has number 1 (so, the human capital has number $N + 1$).

Next M lines are the descriptions of roads and contain three numbers each: the numbers of source and destination towns connected by the road and its capacity. The first two numbers are integers and the third is a real number with at most 4 digits after decimal point. All capacities do not exceed 10^5. Note that for each two corresponding roads, only one of them is given in the input file; you have to reconstruct the road corresponding to it yourself.

Output

On the first line of the output file, write the only number P — the maximal amount of goods that can be transported between the orc and human capitals. On the second line, output M numbers — the amount

of goods transported along every road in a unit of time in the same order in which they are given in the input. Output the amounts as precisely as possible.

Example

transportation.in	transportation.out
4 4 1 2 1 2 3 1 3 8 1 8 6 1	1.0000000000 1.0000000000 0.5000000000 0.5000000000 0.5000000000