Problem A. Ariel (16+)

King Triton really likes watching sport competitions on TV. But much more Triton likes watching live competitions. So Triton decides to set up a swimming competition in the kingdom Merfolk. Thousands of creatures come to take part in competition, that's why it is too difficult to take the first place.

For the King's beloved daughter Ariel this competition is the first in her life. Ariel is very kind, so she wants to give a lot of gold medals. Ariel says, that it is unfair to make a single ranking list for creatures that are so different. It is really a good result to be the fastest small fish without tail in Merfolk!

Ariel chooses k important traits (such as size, tailness, rapacity and so on). A creature can either possess a trait or not (there are no intermediate options).

A *score* is given for each creature (it doesn't matter how it was calculated) and the list of possessed traits $f_1, \ldots, f_y$ is also given.

Ariel wants to know the place occupied by creature a in a competition among creatures, who have the same traits $h_1, \ldots, h_t$. So if creature a doesn't have a trait h_i, then all creatures in the competition are without this trait. If creature a has a trait h_i, then all creatures in the competition have this trait. Other traits doesn't matter. The winner of the competition is a creature with the maximum score.

Input

The first line contains n ($1 \le n \le 10^4$) and k ($1 \le k \le 10$). The next n lines contain information about creatures: *score* ($1 \le score \le 10^9$), y ($0 \le y \le k$) — the number of possessed traits, and y numbers f_i ($1 \le f_i \le k$) — ids of possessed traits. All f_i in one line are different.

The next line contains m ($1 \le m \le 10^5$) — the number of queries from Ariel. The next m lines describe queries: a ($1 \le a \le n$) — the id of a creature, then t — the number of traits, then t numbers h_i. All h_i in one line are different.

Output

For each query output the place of a creature a in ranking list amount the corresponded creatures. If several creatures have the same score all of them take the same place.

Examples

stdin	stdout
3 2 100 1 1 50 1 2 30 2 1 2 12 1 2 1 2 1 1 1 1 1 2 1 0 2 0 2 1 1 2 1 2 2 2 2 1 3 0 3 2 1 2 3 1 2 3 1 1	1 1 1 1 2 1 1 3 1 2 2
3 2 100 0 10 0 100 0 3 1 0 2 0 3 0	1 3 1

Problem B. Billy, Willy and Moscow Underground (21+)

World's economy depression of the 2008 affected everybody in the world and harmed a lot of businesses. But Rich Scrooge McDuck's business suffered most of all. His gold factories started yielding a loss and even went bankrupt. Moreover, the US government revealed that Scrooge was involved in illegal mortgage practices. He had no other option but to leave USA and go abroad. You know what? He chose Russia!

– Uncle Scrooge, we want to try Moscow underground, buy tickets for us! — said Billy and Willy, who had come on their holidays to visit their poor uncle.

– Okey, wait a moment, please — uncle Scrooge answered pensively. He tried to figure out how to spent the least possible money and please his nephews at the same time.

Since Scrooge has never liked mathematics, he needs your help. It is known that Billy and Willy stay in Moscow for n days. They like underground and want to spend some of days to explore it (one trip a day). All the rest days they are going to visit Gorky Park. To prevent squabbles between Billy and Willy, Scrooge needs to give them separate tickets even if they go to underground together. At the end of each day nephews should return their tickets back to uncle. Scrooge has special relations in Moscow underground and can buy tickets for A passages and B days cheaper than they are sold to ordinary people. Certainly, he wants to buy minimum possible number of tickets, so he needs to determine which tickets to give Billy and Willy in the morning. You are hired to help McDuck solve this tricky problem!

Note that the single ticket allows you to use underground no more than A times and the number of days between the first and the last passage should be less than B.

Input

In the first line of the input there is integer n ($1 \leq n \leq 200$) and the integers A and B ($1 \leq A, B \leq 20$). In the second and third lines contain numbers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots b_n$ respectively ($a_i, b_i \in \{0, 1\}$). Billy uses underground on the ith day if and only if $a_i = 1$. Willy uses underground on the ith day if and only if $b_i = 1$.

Output

Output one number — the minimum number of tickets that Scrooge should buy for Billy and Willy.

Examples

stdin	stdout
2 5 5 1 1 0 0	1
2 5 5 1 0 0 1	1
11 20 10 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0	3

Note

In the last test Scrooge should give two tickets at the first day, so they stop working at the eleventh day and he should give Billy one more ticket.

Problem C. Cinderella (6+)

Cinderella is given a task by her Stepmother before she is allowed to go to the Ball.

There are N ($1 \leq N \leq 1000$) bottles with water in the kitchen. Each bottle contains L_i ($0 \leq L_i \leq 10^6$) ounces of water and the maximum capacity of each is 10^9 ounces. To complete the task Cinderella has to pour the water between the bottles to fill them at equal measure.

Cinderella asks Fairy godmother to help her. At each turn Cinderella points out one of the bottles. This is the source bottle. Then she selects any number of other bottles and for each bottle specifies the amount of water to be poured from the source bottle to it. Then Fairy godmother performs the transfusion instantly.

Please calculate how many turns Cinderella needs to complete the Stepmother's task.

Input

The first line of input contains an integer number N ($1 \leq N \leq 1000$) — the total number of bottles.

On the next line integer numbers L_i are contained ($0 \leq L_i \leq 10^6$) — the initial amount of water contained in i^{th} bottle.

Output

Output a single line with an integer S — the minimal number of turns Cinderella needs to complete her task.

Examples

stdin	stdout
3 5 7 7	2
3 21 10 2012	1
1 100	0

Problem D. Darkwing Duck (18+)

Daring duck of mystery, Champion of right, Swoops out of the shadows, Darkwing owns the night. Somewhere some villain schemes, But his number's up.

Cloud of smoke and he appears, Master of surprise. Who's that cunning mind behind That shadowy disguise? Nobody knows for sure, But bad guys are out of luck.

'Cause here comes Darkwing Duck!

And Darkwing Duck needs your help, brother. Sometimes it's very difficult to appear out of cloud of smoke exactly at place where bad guys are doing their bad business. Just imagine how difficult it is if Megavolt plans to attack the east side and Mr Banana Brain will surely appear in the west. The only thing we know is that villains always plan to do the max damage and we should decide what to do to stop them. City consists of some buildings in a line going from the east to the west. Darkwing Duck knows that the bad guys appear somewhere in the segment from l^{th} to r^{th} building, then move to the west destroying everything on their way and disappear somewhere on this segment. Intending to cause max damage for the first victim they always choose the most valuable building, if there are multiple choices, they choose the one with the most valuable next building to destroy, and so on, if there is still a tie, they try to destroy the maximum number of buildings. Help Darkwing Duck to find the exact place of the villains attack.

Input

In the first line of input, there is a string consisting of English letters, digits and symbols ",", "!", "_", ".", "–" representing the town from the east to the west. Length of the string is no more than $500\,000$ symbols. Value of the building is equal to ASCII code of symbol representing it! In the second line there is N ($1 \leqslant N \leqslant 500\,000$) — the number of queries (villains attacks). In the next N lines of input there are numbers $1 \leqslant l_i \leqslant r_i$ — segment of the possible villains attack (it's guarantied that it is completely inside the string). Buildings are numbered starting from 1.

Output

For every query you should print one number on the line: the starting point of the villains attack. And let the force be with you!

Examples

stdin	stdout
Lets-get-dangerous!	17
3	3
1 19	17
3 8	
15 18	

Problem E. Epic Fail of a Genie (18+)

Aladdin had found a new shiny lamp and has started polishing it with his hands. Suddenly a mysterious genie appeared from within and offered Aladdin to fulfill any of his three wishes. Genie had a very subtle humor that made Aladdin very sceptical about him. Aladdin didn't believe that genie was so powerful that could do anything he had wished and asked him to become a mouse. The genie did that without hesitation. Then Aladdin asked genie to become a mouse pad. Genie didn't like this kind of wish but had to submit. Finally Aladdin tested genie's abilities in math: he had to choose a nonempty subset giving the maximum product from the given set of numbers. Genie was shocked. Math was his Achilles' heel, however he was able to contact anyone on earth to help him. You are a secret weapon of the genie — help him solve the test and avoid this epic fail. This is the last chance for the genie: he'll be forever jailed in the lamp if his new master doesn't trust him.

Input

The first line of input contains an integer N ($2 \leq N \leq 10^4$) — the cardinality of a set of numbers.

The second line of input contains N floating-point numbers with absolute value not more than 10^6. The fractional part of each number does not contain more than two digits.

Output

The first line of the output should contain a single integer M — the total number of numbers that genie should choose from the set.

The second line of output should contain 1-based indexes of these numbers. Indexes must be sorted in ascending order. If multiple solutions exist please output the one with the minimal subset cardinality. If there are still several suitable solutions output any of them.

Examples

stdin	stdout
7	4
1 3 0 -1 -2 0.5 3	2 4 5 7

Problem F. Flood (16+)

We all know that King Triton doesn't like us and therefore shipwrecks, hurricanes and tsunami do happen. But being bored with the same routine all these years Triton has decided to make a spectacular flood this year.

He chose a small town in a hilly valley not far from the sea. The power of Triton is enough to pour one heavy rain on the hill. He is worried however that water will miss that chosen town due to various river basins and water flows. Triton asks you to help him help calculate the amount of water that reaches the chosen town.

There are water ponds in a hilly valley on the way to the town. Some of them are connected to each other with rivers. If some pond is overfull with water, the water begins to flow evenly to the connected ponds (or to the sea, if there are no connected ponds). Each pond contains some water initially, and the maximum pond capacity is also known. The chosen town is located on the bank of one pond — you should calculate the water level in this pond after all water flow is run out.

Input

On the first line of input integers N and K ($2 \leqslant N \leqslant 10^4, 0 \leqslant K \leqslant 10^5$) are given — the number of water ponds and the number of pond connections respectively.

On the next N lines of input integers P_i and A_i ($0 \leqslant A_i \leqslant P_i \leqslant 10^6$) are given — these are the maximum i^{th} water pond capacity and its initial water level.

On the next K lines of input integers F_j and T_j ($1 \leqslant F_j, T_j \leqslant N, F_j \neq T_j$) are given — they denote a possible river flow connection from F_j to T_j water pond (reverse water flow is not possible). Consider water flow from a pond to be equally distributed between all possible flow connections from that pond. Triton is absolutely sure that there are no cycles in river flows between the ponds, and there are no multiple rivers between any two ponds.

On the last line of input integers X, Y and Z ($1 \leqslant X, Z \leqslant N, 1 \leqslant Y \leqslant 10^6$) are given — the water pond that receives Triton's heavy rain, the amount of water that is added to this pond and the target pond (near the chosen town) to test respectively.

Consider that excessive water flows from a water pond if and only if its capacity is full. If some pond is overfull and no water flows are defined from that pond consider that all excessive water has flown out to the sea.

Output

The first line of the output should contain a single floating-point number L_z — the final water level in the target pond when all water flow is complete. Answers with absolute or relative error less than 10^{-4} are considered correct.

Examples

stdin	stdout
4 4	3.0
10 10	
1 0	
1 0	
10 0	
1 2	
1 3	
2 4	
3 4	
1 5 4	

Problem G. Gadget Hackwrench (16+)

Chip 'n' Dale rescue rangers! But observant viewers know that help is usually required by Chip and Dale themselves. Today you are in the role of cunning Gadget Hackwrench.

So, Chip and Dale are again in the paws of Fat Cat. He doesn't like rodents much and therefore prepared a treacherous test. He is going to put them to a labyrinth and see if they can escape from it. The labyrinth is actually built as a tree where each edge has fixed direction (by definition *tree* is a connected unoriented graph without cycles).

Gadget has intercepted a talk between Fat Cat and his henchmen about future tests. For each test round she knows the exact location where Chip and Dale are to be put by Fat Cat and the location of an exit. Gadget wants to compute whether they will be able to find an exit for each test.

Input

The first line of input contains an integer N ($1 \leqslant N \leqslant 10^5$) — the number of vertices in a graph.

On the next $N - 1$ lines of input directed arcs of the tree are given. On the $(i + 1)^{th}$ line integer numbers a_i and b_i are given ($1 \leqslant a_i, b_i \leqslant N$) denoting an arc from vertex a_i to vertex b_i. It is guaranteed that arcs $a_1, a_2, \ldots, a_{n-1}$ without orientation form a tree.

Then a string with integer number M ($1 \leqslant M \leqslant 10^5$) is given — the number of queries to process. Next M lines describe queries: $(n + 1 + i)^{th}$ line contain integers x_i and y_i ($1 \leqslant x_i, y_i \leqslant N$).

Output

For each query please output a separate line containing 'Yes' (without quotes) if graph contains a path between x_i and y_i, or 'No' (without quotes) in other case.

Examples

stdin	stdout
4	Yes
1 2	Yes
3 1	No
4 1	Yes
6	No
1 2	No
3 2	
2 3	
4 2	
4 3	
2 1	

Problem H. Hunt for Treasure! (16+)

Scrooge McDuck obtained a plan of the ancient Mayaduck labyrinth full of gold and other treasures. He knows that there are exactly n rooms in the labyrinth and he would like to visit all of them to take all fortune. Every room has exactly one exit to some other room. Every room has no more than one entry from some other room. The main problem is that there are no ways between labyrinth and outer space, and between some pairs of rooms. Scrooge decided to use Duck Teleportation Company Unit to resolve this case. He can teleport to every room inside labyrinth or outside. But there is another problem: every jump costs a huge amount of money and McDuck needs to minimize the number of teleportations.

On the Scrooge's plan rooms are numerated from 1 to N and in every room there is a written room number where one could go to. Unfortunately some numbers are damaged by age and are now unreadable.

You should find the expected amount of money Scrooge needs to investigate all rooms and come back out of the labyrinth, provided that his strategy is optimal and all possible configurations of labyrinth are equiprobable. He begins his journey from outside of the labyrinth.

Input

On the first line of input an integer number T $(0 < T \leqslant 100)$ is given — the total number of testcases. Every testcase is described by two lines. On the first line there are two integers $1 < N \leqslant 4000$ — the number of rooms and $0 \leqslant c \leqslant 1000$ — the cost of one jump. On the second line there are N numbers, $0 \leqslant a_i \leqslant N$ — room number one can go directly from i^{th} room. If any room number is unreadable then $a_i = 0$. All testcases are separated by an empty line. The sum of N in all cases doesn't exceed 4000. It is guaranteed that all input cases are correct, i.e. there exists at least one plan corresponding to the input data.

Output

For every input case you should output one floating point number on a separate line — the answer for the problem with absolute or relative precision of 10^{-6}.

Examples

stdin	stdout
3	1.000000
2 1	2.000000
0 0	1.333333
3 2	
2 3 1	
4 1	
0 0 0 0	

Problem I. Innovative Business (12+)

Chip 'n' Dale have started a new business in the forest: they produce tiles of fixed rectangular size and pave roads with them.

Road paving rules are the following. Starting from one corner of the rectangular road tiles are paved side-by-side without gaps or overlaps. Tiles can be cut into pieces to be used to pave the road if and only if the whole tile doesn't fit. Each tile contains a pattern with parallel lines that must be retained on the paved road. This makes their orientation significant: any tile or its piece can not be rotated. All tile connection lines are straight, parallel to one of the road edges and either perpendicular or parallel to each other. Chip 'n' Dale always pave the road so that each edge of a tile is adjacent to not more than one other tile, and they always pave the road with the least possible amount of tile pieces on the road.

Given the size of the road and the size of one tile please help Chip 'n' Dale determine the number of tiles they need to produce to fully pave the road.

Input

On the first line of input integers $Width_{road}$ and $Length_{road}$ ($1 \leqslant Width_{road}, Length_{road} \leqslant 10\,000$) are given — the width and the length of the road respectively.

On the second line of input integers $Width_{tile}$ and $Length_{tile}$ ($1 \leqslant Width_{tile} \leqslant Width_{road}$, $1 \leqslant Length_{tile} \leqslant Length_{road}$) are given — the width and the length of the tile respectively.

Output

The first line of the output should contain a single integer number N — the minimal number of whole tiles needed to fully pave the road according to Chip 'n' Dale road paving rules.

Examples

stdin	stdout
10 10 2 2	25
3 5 2 2	4
35 17 25 1	26

Problem J. Just Another Disney Problem (14+)

Evil wizard Jafar has a huge collection of lamps. He likes touching them, wiping the dust, looking at his reflection in them. He loves all of them almost equally, they are very beautiful, but for every two of them he likes one of them more than another one. Jafar keeps his lamps in a very long hall, all lamps in one line. One day he decided to arrange all lamps along his way from one side of the hall to another in such a way, that for every two neighboring lamps he likes the next one more than previous. In other words Jafar would like a some kind of an ascending order of lamp's quality. You are a new servant of wizard and you should fulfill the desire of your master. The main problem is that you don't know anything about Jafar's preferences. You can ask Jafar for any two lamps which one is better, but you should be careful, he is very busy now in his plans for world domination and you shouldn't ask him too much questions. Note that preferences could be non-transitive. You should output desired order of all lapms or report Jafar that it doesn't exist.

Input

First number — N ($1 \leqslant N \leqslant 1000$). Answer for every question — string "YES", if Y is better than X, and "NO", if X is better than Y.

Output

Your questions — one line with three integers 1, X, Y ($1 \leqslant X, Y \leqslant N$, $X \neq Y$). You should ask not more than $10\,000$ questions. In the last line: integer 0, then N integers a_i ($1 \leqslant a_i \leqslant N$) — the desired permutation or N zeros if such permutation doesn't exist. All integers in the lines should be separated by spaces.

Examples

stdin	stdout
3	1 1 2
YES	1 2 3
NO	1 1 3
NO	0 3 1 2

Note

The pipe from your program to the interactor program and the pipe back have limited size. Your program must read from the standard input to avoid deadlock. Deadlock condition is reported as Wall time-limit exceeded.

To flush the standard output stream use the following statements:

In C use fflush(stdout);

In C++ use cout.flush();

In Java use System.out.flush();

If your program receives EOF (end-of-file) condition on the standard input, it MUST exit immediately with exit code 0. Failure to comply with this requirement may result in Time-limit exceeded error.

Problem K. Key to Magica's diary (14+)

He's smart, he's rich — and he needs your help. Scrooge McDuck's Number One Dime is stolen by Magica De Spell. Best Scotland Yard inspectors are working on this case but the only evidence they found is Magica's accidentally dropped secret diary. Unfortunately, it is protected by some spell that prevents everybody from understanding its contents. That's why they need you - the best mathematician of Duckburg.

The first thing you noticed is that some words are occurring much more often than others and form some weird pattern. You've made a hypothesis that these words are part of Magica's spell and decided to ignore them for future analysis. After that it became pretty obvious that some words in the diary are found next to each other quite often (such as 'Scrooge' and 'McDuck') while others will be located pretty much independent from each other. You believe that this is the key to Magica's diary and want to check your idea.

More precisely, word is a non-empty continuous sequence of English letters. Text may contain other symbols, you should consider them as spaces. Words are considered to be case-insensitive. Define $P(a)$ as percentage of word a in the text (i.e. the number of a occurrences divided by the total number of words in the text). Similarly, you define $P(a, b)$ as the rate of words a and b occurrences adjacent to each other (i.e. number of such occurrences divided by total number of adjacent word pairs in the text). Then

$$C(a, b) = \frac{P(a, b)}{P(a)P(b)}$$

will show you how dependent words a and b are.

You are particularly interested in some pairs of words and want to check how dependent they are. Unfortunately, the diary is too large to do this analysis manually, so you decided to write a program to do that for you.

Input

First line of input contains one integer number N — the total number of lines in Magica's diary ($1 \leqslant N \leqslant 4000$). Then N lines of text follow. Diary contains only English letters, brackets, punctuation marks (`0123456789.,:;-!?'()"`), ampersands and spaces. The total length of the text is less or equal than 500KiB, and the length of each word is not more than 20. The text also contains more than one non-magic word. $N + 2^{nd}$ line contains K — the total number of 'magic' words that should be ignored ($0 \leqslant K \leqslant 100$). Next K lines contain these words, one per line, lowercase. $N + K + 3^{rd}$ line contains Q — the total number of word pairs you're interested in ($0 \leqslant Q \leqslant 50\,000$). Then Q lines follow, each containing two lowercase words.

Output

For each query (a, b) you should output the value of $C(a, b)$ as a floating point number on the separate line with absolute or relative precision of 10^{-6}.

Examples

stdin	stdout
3 I have lived through some terrible things in my life, some of which actually happened. 3 of in which 3 some actually things life lived through	6.545454545454546 0.0 13.090909090909092
1 (2 - 2) Plus (1 - 1) equals 0 0 1 equals plus	4.0