

Problem A. The Power of the Dark Side

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

On the Jedi Tournament n Jedi are battling each other. Every Jedi has three parameters: strength, agility and intelligence. All parameters of all Jedi are different. When two Jedi have a fight, the winner will be the Jedi which has more parameters higher compared to the same parameters of his opponent. For example, Jedi with parameters (5, 9, 10) defeats Jedi with parameters (2, 12, 4) because of first and third parameters.

Sith have a plan to turn one the Jedi to the dark side of the Force. It will give him a powerful skill that allows to swap some his parameters, maybe more than one time, before every fight. Sith wish their new apprentice to defeat all of the remaining Jedi.

Which Jedi can be used by Sith to lead their plan to success?

Input

The first line contains the only integer n ($1 \leq n \leq 200000$) — the number of the Jedi.

Then n lines follow. Each of them contains three integers separated by space: a_i , b_i and c_i ($1 \leq a_i, b_i, c_i \leq 10^9$) — the parameters of the Jedi.

All a_i , b_i and c_i are pairwise distinct.

Output

In the first line output one integer m — the number of the suitable Jedi.

In the second line output m integers — the numbers of these Jedi — in the ascending order.

Examples

stdin	stdout
4 5 9 10 2 12 4 8 7 3 6 11 1	2 1 4
4 4 1 11 5 8 9 2 12 6 7 3 10	3 2 3 4

Problem B. Similar Strings

Input file: **stdin**
Output file: **stdout**
Time limit: 3 seconds
Memory limit: 256 megabytes

Let's call two strings *similar* if there exists a bijective mapping over characters, which, when applied to the characters of the first string, makes it equal to the second string. For example, «**abacaba**» and «**tetatet**» are similar strings, but «**test**» and «**bear**» — are not. Given the set of strings, find the number of pairs of similar strings.

Input

The first line contains one integer n — the number of strings.

Next n lines contain non-empty strings of lowercase Latin letters. The sum of lengths of these strings does not exceed 10^6 .

Output

Output the only integer —the number of pairs of similar strings.

Examples

stdin	stdout
4 abacaba tetatet test bear	1
4 jury code will pass	2
4 your code wont pass	3

Problem C. Victor's Research

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

Unacknowledged scientist Victor conducts a pseudoscientific research of the relation between integers that cross his mind and the integer that comes into his assistant's mind. He wrote the integers a_1, \dots, a_n which had crossed his mind. Then it turned up that the integer s had come into his assistant's mind. Victor wants to determine how many consecutive non-empty sets of integers a_l, a_{l+1}, \dots, a_r ($l \leq r$) have the sum $a_l + a_{l+1} + \dots + a_r = s$.

Input

The first line contains two integers separated by space: n and s ($1 \leq n \leq 200000$, $-2 \cdot 10^{14} \leq s \leq 2 \cdot 10^{14}$) — the number of integers which crossed Victor's mind and the integer that came into his assistant's mind.

The second line contains n integers separated by space: a_i ($-10^9 \leq a_i \leq 10^9$) — the integers which crossed Victor's mind.

Output

Output the only integer — the number of consecutive non-empty sets of integers which have the sum s .

Examples

stdin	stdout
5 2 -1 1 2 -1 1	5
6 3 3 -2 1 -1 1 2	3

Problem D. Hamming Distance

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

Once the boy named Richard received the cat as a gift. Of course, he immediately started to think up how to name him. Three his friends offered their variants: strings a , b and c of equal lengths n . Richard notes the distance $\rho(s_1, s_2)$ between strings s_1 and s_2 as a number of positions in which the characters in these strings differ. For example, $\rho(\text{dark}, \text{duck}) = 2$, $\rho(\text{beer}, \text{wine}) = 4$. Richard wants to please his friends and name the cat such the name s that $\rho(s, a) + \rho(s, b) + \rho(s, c)$ is minimal.

Input

The first line contains one integer n ($1 \leq n \leq 200000$) — the length of the strings.

Then three strings a , b и c of the same length n follow — the names offered by Richard's friends. The strings consist of the lowercase Latin letters.

Output

Output the only string of the length n , consisting of the lowercase Latin letters — the name for the cat. If there are many possible names satisfying the condition, output any of them.

Examples

stdin	stdout
6 needle turkey bottle	turtle

Problem E. Of Groups and Rights

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

Computer security specialist Alexey is setting up the access rights for the secret information about a certain programming contest. Security system is organized so that users are united into the groups numbered from 1 to n . Some groups can include the others. Each group can be directly included only in one other group, but one group can include several other groups. Group 1 includes all other groups directly or indirectly. If the group does not include any other group, then it's actually a user. And vice versa, if the group is a user, it can't include any other groups.

For each group the access to the secret information can be allowed, denied or not set. If the access for the group is not set, it's inherited from the group that includes this one. If access is not set for that group too, it's inherited again and so on. If the access is not set even for the group 1, it's considered denied for the specified group.

Alexey has instructions about the groups, the rights for them and how the groups include each other. He wants to optimize his data structure in order that only the information about the allowed rights is stored, and for other groups it's considered not set. At the same time all users' access rights should be the same as followed from the initial instructions. Also Alexey wants new data structure to require less memory, so there should be minimal number of the permissions records. From all such structures he wants to select that one, for which the sum of the nesting levels for all stored groups is minimal.

Input

The first line contains an integer n ($1 \leq n \leq 200000$) — the number of groups.

Then the string of n characters «+», «-», «0» follows. Character «+» on the i -th position means the access is allowed for the i -th group, «-» — that the access is denied, and «0» — that it's not set.

Then $n - 1$ lines follow. Each of them contains two integers separated by a space: a_i and b_i ($1 \leq a_i, b_i \leq n$). Each of these records means that group a_i includes the group b_i .

Output

In the first line output the only integer m — minimal number of the groups, about which allowed access information should be stored.

In the second line output m integers separated by spaces — numbers of these groups in the ascending order.

Examples

stdin	stdout
10 +--+000++-+ 1 2 1 5 1 9 2 6 2 8 5 4 5 10 9 3 9 7	3 5 8 9

Problem F. Battle Fury

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

A hero named Magina is battling the group of n monsters with the help of the legendary axe known as Battle Fury. Each monster has a_i hit-points. Every hit Magina lowers the health of the monster which was directly hit by p hit-points, while the health of all other monsters lowers by q hit-points. If the monster's hit-points become non-positive, this monster dies. Every hit Magina wants to choose the target properly because he's going to kill all monsters in a minimal number of hits. You should determine this number.

Input

The first line contains three integers separated by spaces: n , p and q ($1 \leq n \leq 200000$, $1 \leq q \leq p \leq 10^9$) — the number of monsters, the damage on the target and the damage on the others.

The second line contains n integers separated by spaces: a_i ($1 \leq a_i \leq 10^9$) — monsters' hit-points.

Output

Output the only integer — minimal number of hits that Magina needs to kill all monsters.

Examples

stdin	stdout
2 3 2 5 5	2
3 5 3 17 13 14	5

Problem G. City Square

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

Mayor of one big city Sergey decided to pave the infinitely large central square of the city by the multicolored tiles. All tiles are squares of the same size. They should be placed next to each other, making the square grid. He has managed to get an infinite amount of tiles of n different colors at a reasonable price, and he wants to use all colors.

Sergey decided to create the paving design by himself. He is a perfectionist, so he wants the pattern to satisfy two conditions.

- For every two colors it should be possible to transmute the set of tiles of the first color to the set of tiles of the second color using only translation.
- For every color on the plane such Cartesian coordinate system should exist so that the set of centers of all tiles of this color is the same as the set of integer points in this system.

Now Sergey is lost in thought, if such a design can be realized at all.

Input

The only line contains the only integer n ($1 \leq n \leq 10^9$) — the number of colors of tiles.

Output

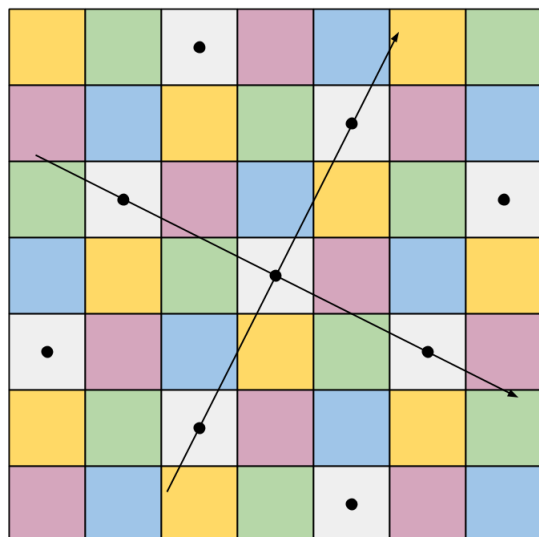
Output «Yes» (without quotes), if Sergey is able to realize his idea, and «No» (without quotes), otherwise.

Examples

stdin	stdout
3	No
4	Yes
5	Yes

Note

The following picture shows one of the ways to pave a plane with tiles of 5 colors.



Problem H. Secret Information

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

Little known hacker Heaven just got access to the secret information about problems of some major programming contest. He wants to change statement of one of the problems to sabotage the contest.

For Heaven, initial and target statements are just strings of zeroes and ones. He can take any substring of initial statement and negate it — replace each zero with one, and each one with zero.

Heaven would like to leave no trace of his actions, so he wants to minimize the number of steps in turning initial statement to target.

Input

The first line contains one integer n ($1 \leq n \leq 200000$) — the length of the strings.

Then two strings of «0» and «1» follow — the binary codes of initial and target statement.

Output

Output one integer — the minimal possible number of steps.

Examples

stdin	stdout
6 101010 110011	2
7 1010101 0011100	3

Problem I. Meteor Flow

Input file: **stdin**
Output file: **stdout**
Time limit: 2 seconds
Memory limit: 256 megabytes

The spaceship «Enterprise» has got into meteor flow and is under threat of destruction. Once this happens, Captain Kirk ordered to switch the force shield on, and onboard computer instantly determined n meteors which give no chance to be evaded. Assuming the force shield is switched on at the moment 0, it is known that i -th meteor will strike the «Enterprise» at the moment t_i and deal d_i units of damage.

Each unit of time the force shield increases its power by 1. When the meteor strikes the «Enterprise», the ship receives damage if the power of the shield is less than the meteor's damage. Otherwise, the power of the force shield decreases by the meteor's damage.

The «Enterprise» has a cannon, each shot of which can knock any meteor off. But Captain Kirk does not like unnecessary firing and wants to overcome the meteor flow without any damage received and doing as few shots as possible.

Input

The first line contains a single integer n ($1 \leq n \leq 200000$) — the number of meteors.

Then n lines follow. Each of them contains two integers separated by space: t_i and d_i ($1 \leq t_i, d_i \leq 10^9$) — the moment of time when i -th meteor will strike the ship and the damage it will deal. All t_i are pairwise distinct and given in ascending order.

Output

Output a single integer — the minimal number of shots that needs to be done in order to not receive any damage.

Examples

stdin	stdout
3 3 2 5 4 6 3	1
5 1 2 3 2 5 3 6 2 7 3	2

Problem J. The Best Statement

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

Team Samara SAU Teddy Bears is making the statement for one problem for the qualification round. Pavel generates the variants of the statement one after another while Alexey for each of these variants says whether he approves the variant or not. The process will be finished just after Alexey will approve one variant of the statement. Nikita does not formally involved in this process but he's rooting for the statement with all his heart.

Nikita understood the principle: for each Pavel's statement there are Alexey's and Nikita's grades of its quality. Nikita figured out Pavel's behavior and guessed that Pavel will generate n statements so that Alexey's grades for them will be a_i and Nikita's ones — b_i . Alexey will approve the statement if he evaluates it greater than or equal to k .

Before all the process, Nikita can predetermine Alexey's mood and set an arbitrary value to k . Then this value will be fixed for the rest of process. Of course, Nikita should select the value of k so that at least one of the variants of the statement will be approved by Alexey. Nikita wants to maximize his own grade of the approved statement. What is the best result he can achieve?

Input

The first line contains a single integer n ($1 \leq n \leq 200000$) — the number of the variants of the statement. Then n lines follow. Each of them contains two integers separated by space: a_i and b_i ($1 \leq a_i, b_i \leq 10^9$) — grades of Alexey and Nikita for the i -th statement, correspondingly.

Output

Output a single integer — the maximal Nikita's grade of the statement he can achieve.

Examples

stdin	stdout
3 3 3 4 6 2 5	6
3 2 5 4 3 3 6	5

Problem K. Three Contests

Input file: `stdin`
Output file: `stdout`
Time limit: 4 seconds
Memory limit: 256 megabytes

Three contests have finished at the programming training camp. Now every team considers itself stronger than every other team which has been beaten by it at least at the one of these contests.

How many pairs of teams, where each team considers itself stronger than the other, exist?

Input

The first line contains the only integer n ($1 \leq n \leq 200000$) — the number of teams at the training camp.

Each of the next n lines contains three integers: a_i , b_i and c_i ($1 \leq a_i, b_i, c_i \leq n$) — the places taken by team i at the first, the second and the third contest correspondingly.

It's guaranteed that there is no contest where some teams share the same place, that is, all a_i are distinct, all b_i are distinct, and all c_i are distinct.

Output

Output the only integer — number of pairs of teams where each team considers itself stronger than the other.

Examples

stdin	stdout
4 1 3 1 2 2 4 4 1 2 3 4 3	5

Problem L. For the Honest Election

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

The presidential election is held in the state R and n citizens have the vote. Candidate P has not really enough supporters, but he has a friend in the election committee, so he proposed a new voting scheme. The election committee has two options.

In the first option each citizen votes for the one of the citizens he trusts. The citizen who has more votes than any other citizen chooses any citizen he wants as a president. Of course if the supporter of the candidate P is that elected citizen, P will be a president.

In the second option the committee divides the voters into the few groups of the same size, and the representative in every group is elected using the same scheme. That means the group can be divided again or the election described in the first option can be held. After the representatives for each group are elected they choose the representative among each other using the first option.

Candidate P is planning to use his friend from the election committee to divide the citizens into the groups by himself. Note that the supporters of the candidate P are absolutely honest with each other and for the common good they can arrange whom to vote for in each election. Now P thinks about the minimal number of supporters he should have to win the election for sure.

Input

The only line contains one integer n ($1 \leq n \leq 10^9$) — the number of the citizens of R who has the vote.

Output

Output the only integer — minimal number of the supporters candidate P needs to win the election for sure.

Examples

stdin	stdout
9	4
12	6

Note

Let's consider the first sample. P can divide 9 voters into 3 groups of 3 people. Inside each group he needs 2 votes for the loyal representative election. At the same time only 2 of 3 groups are enough for him to elect the loyal representative. Therefore he should send 2 supporters in the first group and other 2 in the second one to win the election.