Codeforces Beta Round #19

A. World Football Cup

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

Everyone knows that 2010 FIFA World Cup is being held in South Africa now. By the decision of BFA (Berland's Football Association) next World Cup will be held in Berland. BFA took the decision to change some World Cup regulations:

- the final tournament features n teams (n is always even)
- the first $n / 2$ teams (according to the standings) come through to the knockout stage
- the standings are made on the following principle: for a victory a team gets 3 points, for a draw — 1 point, for a defeat — 0 points. In the first place, teams are ordered in the standings in decreasing order of their points; in the second place — in decreasing order of the difference between scored and missed goals; in the third place — in the decreasing order of scored goals
- it's written in Berland's Constitution that the previous regulation helps to order the teams without ambiguity.

You are asked to write a program that, by the given list of the competing teams and the results of all the matches, will find the list of teams that managed to get through to the knockout stage.

Input

The first input line contains the only integer n ($1 \le n \le 50$) — amount of the teams, taking part in the final tournament of World Cup. The following n lines contain the names of these teams, a name is a string of lower-case and upper-case Latin letters, its length doesn't exceed 30 characters. The following $n \cdot (n - 1) / 2$ lines describe the held matches in the format `name1-name2 num1:num2`, where $name1$, $name2$ — names of the teams; $num1$, $num2$ ($0 \le num1, num2 \le 100$) — amount of the goals, scored by the corresponding teams. Accuracy of the descriptions is guaranteed: there are no two team names coinciding accurate to the letters' case; there is no match, where a team plays with itself; each match is met in the descriptions only once.

Output

Output $n / 2$ lines — names of the teams, which managed to get through to the knockout stage in lexicographical order. Output each name in a separate line. No odd characters (including spaces) are allowed. It's guaranteed that the described regulations help to order the teams without ambiguity.

Examples

input
4 A B C D A-B 1:1 A-C 2:2 A-D 1:0 B-C 1:0 B-D 0:3 C-D 0:3
output
A D

input
2 a A a-A 2:1
output
a

B. Checkout Assistant

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bob came to a cash & carry store, put n items into his trolley, and went to the checkout counter to pay. Each item is described by its price c_i and time t_i in seconds that a checkout assistant spends on this item. While the checkout assistant is occupied with some item, Bob can steal some other items from his trolley. To steal one item Bob needs exactly 1 second. What is the minimum amount of money that Bob will have to pay to the checkout assistant? Remember, please, that it is Bob, who determines the order of items for the checkout assistant.

Input

The first input line contains number n ($1 \le n \le 2000$). In each of the following n lines each item is described by a pair of numbers t_i, c_i ($0 \le t_i \le 2000, 1 \le c_i \le 10^9$). If t_i is 0, Bob won't be able to steal anything, while the checkout assistant is occupied with item i.

Output

Output one number — answer to the problem: what is the minimum amount of money that Bob will have to pay.

Examples

input
4 2 10 0 20 1 5 1 3
output
8

input
3 0 1 0 10 0 100
output
111

C. Deletion of Repeats

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Once Bob saw a string. It contained so many different letters, that the letters were marked by numbers, but at the same time each letter could be met in the string at most 10 times. Bob didn't like that string, because it contained repeats: a repeat of length x is such a substring of length $2x$, that its first half coincides character by character with its second half. Bob started deleting all the repeats from the string. He does it as follows: while it's possible, Bob takes the shortest repeat, if it is not unique, he takes the leftmost one, and deletes its left half and everything that is to the left of this repeat.

You're given the string seen by Bob. Find out, what it will look like after Bob deletes all the repeats in the way described above.

Input

The first input line contains integer n ($1 \le n \le 10^5$) — length of the string. The following line contains n space-separated integer numbers from 0 to 10^9 inclusive — numbers that stand for the letters of the string. It's guaranteed that each letter can be met in the string at most 10 times.

Output

In the first line output the length of the string's part, left after Bob's deletions. In the second line output all the letters (separated by a space) of the string, left after Bob deleted all the repeats in the described way.

Examples

input
6 1 2 3 1 2 3
output
3 1 2 3

input
7 4 5 6 5 6 7 7
output
1 7

D. Points

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pete and Bob invented a new interesting game. Bob takes a sheet of paper and locates a Cartesian coordinate system on it as follows: point $(0, 0)$ is located in the bottom-left corner, Ox axis is directed right, Oy axis is directed up. Pete gives Bob requests of three types:

- `add x y` — on the sheet of paper Bob marks a point with coordinates (x, y). For each request of this type it's guaranteed that point (x, y) is not yet marked on Bob's sheet at the time of the request.
- `remove x y` — on the sheet of paper Bob erases the previously marked point with coordinates (x, y). For each request of this type it's guaranteed that point (x, y) is already marked on Bob's sheet at the time of the request.
- `find x y` — on the sheet of paper Bob finds all the marked points, lying strictly above and strictly to the right of point (x, y). Among these points Bob chooses the leftmost one, if it is not unique, he chooses the bottommost one, and gives its coordinates to Pete.

Bob managed to answer the requests, when they were 10, 100 or 1000, but when their amount grew up to $2 \cdot 10^5$, Bob failed to cope. Now he needs a program that will answer all Pete's requests. Help Bob, please!

Input

The first input line contains number n ($1 \le n \le 2 \cdot 10^5$) — amount of requests. Then there follow n lines — descriptions of the requests. `add x y` describes the request to add a point, `remove x y` — the request to erase a point, `find x y` — the request to find the bottom-left point. All the coordinates in the input file are non-negative and don't exceed 10^9.

Output

For each request of type `find x y` output in a separate line the answer to it — coordinates of the bottommost among the leftmost marked points, lying strictly above and to the right of point (x, y). If there are no points strictly above and to the right of point (x, y), output `-1`.

Examples

input
7 add 1 1 add 3 4 find 0 0 remove 1 1 find 0 0 add 1 1 find 0 0
output
1 1 3 4 1 1

input
13 add 5 5 add 5 6 add 5 7 add 6 5 add 6 6 add 6 7 add 7 5 add 7 6 add 7 7 find 6 6 remove 7 7 find 6 6 find 4 4
output
7 7 -1 5 5

E. Fairy

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Once upon a time there lived a good fairy A. One day a fine young man B came to her and asked to predict his future. The fairy looked into her magic ball and said that soon the fine young man will meet the most beautiful princess ever and will marry her. Then she drew on a sheet of paper n points and joined some of them with segments, each of the segments starts in some point and ends in some other point. Having drawn that picture, she asked the young man to erase one of the segments from the sheet. Then she tries to colour each point red or blue so, that there is no segment having points of the same colour as its ends. If she manages to do so, the prediction will come true. B wants to meet the most beautiful princess, that's why he asks you to help him. Find all the segments that will help him to meet the princess.

Input

The first input line contains two integer numbers: n — amount of the drawn points and m — amount of the drawn segments ($1 \le n \le 10^4, 0 \le m \le 10^4$). The following m lines contain the descriptions of the segments. Each description contains two different space-separated integer numbers v, u ($1 \le v \le n, 1 \le u \le n$) — indexes of the points, joined by this segment. No segment is met in the description twice.

Output

In the first line output number k — amount of the segments in the answer. In the second line output k space-separated numbers — indexes of these segments in ascending order. Each index should be output only once. Segments are numbered from 1 in the input order.

Examples

input
4 4 1 2 1 3 2 4 3 4
output
4 1 2 3 4

input
4 5 1 2 2 3 3 4 4 1 1 3
output
1 5