Problem A: Kings

(blue balloon)

Mahya loves to know more about the history of her country. She is in particular interested in the history of the ancient kings of Persia. Recently, Mahya got curious to know how long each of her favorite kings had lived. So, she started searching the web, and collecting information about the kings lives.

Unfortunately, in most cases, the exact dates on which the kings were born or died are not available in resources. So, Mahya could only find some ranges for possible dates of birth and death for each of the kings. For example, for Cyrus the Great, she could only find that the date of birth was between 600 BC and 575 BC, and the date of death was 530 BC. So, she concluded that Cyrus the Great had lived at least 45 years and at most 70 years.

Mahya has created a long list of her favorite kings, and for each king, has written down two ranges showing the birth range and the death range of that king. Since the list is a bit lengthy, she needs your help to process the list, and produce for each king the minimum and the maximum age. Note that if a king was born in year x and died in year y, then he lived y x years.

Standard Input

There are multiple test cases in the input. Each test case consists of a line containing four integers a, b, c, d, where $5000 \leqslant a \leqslant b < c \leqslant d \leqslant 2000$. The range $[a, b]$ shows the birth range, and $[c, d]$ shows the death range. The input terminates with `0 0 0 0` which should not be processed.

Standard Output

For each test case, output a line containing the minimum and the maximum age as two integers separated by a space.

Sample Input	Sample Output
100 110 180 185 -600 -575 -530 -530 -25 10 72 86 0 0 0 0	70 85 45 70 62 111

Problem B: HTML color

(red balloon)

Web colors are colors used in displaying web pages. Each color may be specified either as an RGB triple, or a common English name used for that color. Colors are specified according to the intensity of their red, green and blue components, each repre- sented by eight bits. Thus, there are 24 bits used to specify a web color, and totally 16,777,216 colors can be imagined as web colors. But the HTML 4 specification defines only 16 named colors as shown in the table.

#	Name	Red	Green	Blue
1	White	255	255	255
2	Silver	192	192	192
3	Gray	128	128	128
4	Black	0	0	0
5	Red	255	0	0
6	Maroon	128	0	0
7	Yellow	255	255	0
8	Olive	128	128	0
9	Lime	0	255	0
10	Green	0	128	0
11	Aqua	0	255	255
12	Teal	0	128	128
13	Blue	0	0	255
14	Navy	0	0	128
15	Fuchsia	255	0	255
16	Purple	128	0	128

It is often useful to map one given color to one of the HTML named colors. The goal of this problem is to perform just such a mapping in the RGB color space. The input to the program consists of a collection of RGB color values to be mapped to the closest HTML named color.

For a given color, the "closest" color in the HTML color names is a color with the smallest Euclidean distance from the given color. That is, if *rgb* is the color to be mapped, and $R_1G_1B_1, \ldots, R_{16}G_{16}B_{16}$ is the set of the HTML colors, the closest color is the one which minimizes the distance expression

$$d = \sqrt{(R_i - r)^2 + (G_i - g)^2 + (B_i - b)^2}$$

where i is an integer from 1 to 16.

Standard Input

There are multiple test cases in the input. Each test case consists of a line containing three integers $0 \leqslant r, g, b \leqslant 255$ which are the Red, Green and Blue intensities of the color, respectively. The input terminates with `-1 -1 -1` which should not be processed.

Standard Output

For each test case, output a line containing the name of the closest HTML color to the given color. If there are more than one closest color, print the one which has a smaller associated number in the above table.

Sample Input	Sample Output
120 120 10 111 112 113 5 135 8 -1 -1 -1	Olive Gray Green

Problem C: Scoreboard

(yellow balloon)

Professor Boffin is a regional contest director of ACM ICPC. He loves watching and analyzing the scoreboard during the contest. He believes that the scoreboard is *ideal* when all these criteria hold together:

- Each team has solved at least one problem.
- No team has solved all the problems.
- Each problem is solved by at least one team.
- No problem is solved by all the teams.

Obviously, the scoreboard is not ideal at the beginning of the contest, but it may become ideal during the contest. The scoreboard may remain ideal through the end of the contest, or it may lose this property some time later during the contest. In the latter case, it can be shown that it will never become ideal any more. Given the list of the submissions in a regional contest, you must determine the interval in which the scoreboard was ideal.

Standard Input

The input consists of several test cases. Each test case starts with a line containing 3 space-separated integers T , P , and S which represent the number of teams, problems, and submissions respectively ($1 \leqslant T \leqslant 150, 1 \leqslant P \leqslant 15, 0 \leqslant S \leqslant 5000$). Each of the next S lines represents a contest submission with 4 space-separated parameters:

- teamID: the identifier of the team, an integer in the range [1..T].
- problemID: the identifier of the problem, an uppercase letter from the first P letters of English alphabet.
- submission-time: the time of submission, in HH:MM:SS format, all 3 parts are exactly 2 digits (00 ⩽ HH ⩽ 05, 00 ⩽ MM, SS ⩽ 59).
- result: the result of the submission. It can be one of the following sentences:
    - Yes: Only this case shows that the corresponding team has successfully solved the problem.
    - No - Compilation Error: Unsuccessful submission due to a compilation error in the submitted program.
    - No - Wrong Answer: Unsuccessful submission since the submitted program had a wrong output.
    - No - Run-time Error: Unsuccessful submission due to a run-time error during the execution of

the submitted program.

- No - Time Limit Exceeded: Unsuccessful submission since the execution of the submitted program did not finish in the time limit.
- No - Presentation Error: Unsuccessful submission due to a formatting error in the output of the submitted program.

No two submissions have the same time. The input terminates with a line containing `0 0 0` which should not be processed as a test case.

Standard Output

For each test case, output a line containing the ideal-interval of the corresponding contest. The interval must be provided with two times in exact HH:MM:SS format (as described in the input). The first time shows the moment the scoreboard becomes ideal, and the second time shows the moment the scoreboard is not ideal anymore. If the scoreboard remains ideal through the end of the contest, the second time must be `--:--:--`. If the scoreboard never becomes ideal throughout the contest, both times must be `--:--:--`.

Sample Input	Sample Output
2 3 5 1 A 00:10:05 Yes 2 A 00:15:15 No - Wrong Answer 1 C 01:01:01 Yes 2 B 02:20:00 Yes 1 B 03:10:00 Yes 2 3 5 1 A 00:10:05 Yes 2 A 00:15:15 No - Wrong Answer 1 C 01:01:01 Yes 2 B 02:20:00 Yes 1 B 03:10:00 No - Wrong Answer 2 3 5 1 A 00:10:05 Yes 1 C 01:01:01 Yes 2 A 00:15:15 No - Wrong Answer 1 B 03:10:00 Yes 2 B 04:20:00 Yes 0 0 0	02:20:00 03:10:00 02:20:00 --:--:-- --:--:-- --:--:--

Problem D: Floor
(Orange balloon)

The new building for the Computer Engineering department has several elevators, but has no stairs. In order to facilitate access to lecture rooms and offices, the manufacturer has set each elevator to stop only at certain pre-defined floors; like some elevators would only stop at odd floors and some only at even floors. But, things are more complicated than this and the buttons inside and outside of each elevator would only work for the set of floors that elevator is assigned to stop at. This has made some improvements in reaching certain destination floors, especially for the faculty members, but has caused a lot of confusions for people like students. If a person p is at floor i and wants to go to floor j, which elevator should p take, and to which elevator(s) and on which floor(s) should (s)he transfer to, so that p reaches floor j with minimum travel time? We define p's travel path as $i = f_1 \rightarrow f_2 \rightarrow \cdots \rightarrow f_k = j$, then p's travel time that we want to minimize is $\sum_{r=1}^{k-1} |f_i - f_{i+1}|$. You have been asked to write an application to help people using these elevators.

Standard Input

There are multiple test cases in the input. The first line of each test case specifies n ($1 \leqslant n \leqslant 10$), the number of elevators, followed by the source and destination floors. The i-th line ($1 \leqslant i \leqslant n$) of the next n lines starts with m_i ($2 \leqslant m_i \leqslant 150$), the number of floors at which the i-th elevator may stop, followed by a list of m_i non-negative floor numbers (all numbers are less than 150). The input terminates with a line containing `0 0 0` which should not be processed.

Standard Output

For each test case, output a line containing the minimum travel time that one needs to reach the destination floor when starting from the source floor. It is guaranteed that there is always a possible way from the source floor to the destination floor.

Sample Input	Sample Output
2 2 5 5 0 1 3 5 7 5 0 2 4 6 8 3 3 8 6 0 1 2 3 4 5 5 0 6 7 8 9 4 0 4 5 6 0 0 0	7 5

Problem E : Network

(purple balloon)

After a fierce battle with his opponent, Bruce Wayne finally won the elections and became the mayor of Gotham. Like every other politician, he had an agenda with lots of projects for the sake of Gotham's prosperity, but he was met with the same problem, lack of fund.

He decided to tackle the problem from a different perspective; he will allow companies to buy roads in the city (roads in the city are undirected). The city will get the money needed for the projects and the companies can use the roads for advertisements (or so he thought).

After the deal was done, the companies were more cunning than he expected. They started to threaten that they will block exactly one road in the city and prevent people from getting to their work, in the hope that people will revolt against Mayor Wayne. The problem was that the city is designed as a tree of connected zones, where there is only one unique path between any two zones. Hence, blocking a road means that some zones are not reachable from others anymore.

Mayor Wayne discussed the problem with his council and identified what they called vulnerable roads. A road is vulnerable if blocking it can disconnect two zones from each other. Mayor Wayne wants to prevent this from happening by building more roads but his budget could afford building only one extra road. Can you help him figure out which road he should build, such that he minimizes the number of vulnerable roads?

Standard Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer T, the number of test cases ($1 \le T \le 100$) followed by T test cases. The first line of each test case will contain one integer N, the number of zones in the city ($1 \le N \le 10,000$). The following N - 1 lines will each contain a pair of integers x and y separated by a single space ($1 \le x, y \le N$) which means that zone x is connected to zone y. It's guaranteed that the edges will form a tree.

Standard Output

For each test case, print a single line containing an integer, the minimum number of vulnerable roads in the city after building the new road.

Sample Input	Sample Output
2 3 1 2 1 3 4 1 2 2 3 2 4	0 1