Problem F: Departures

Time limit: 8s, memory limit: 1GB.

At the station the weekly departures board has broken. It does not display where the trains are going. However, it correctly displays the departure time and the travel time of n trains leaving the station every week. You know that there is exactly one single-track route to each destination. This means that trains going in the same direction cannot overtake. Moreover, the routes leading to different cities are separate.

Knowing the timetable, determine d – the smallest number of destinations, so that the trains run on schedule – and group the trains into d groups so that all trains in the group can be assigned to the same destination city.

Take into account that according to the rail transportation policy:

- the trains leaving at the same time may enter the same route in any order;
- the trains run according to the timetable, in particular they do not arrive at the destination station prematurely, i.e. they spend the entire travel time on the route;
- several trains may arrive at the destination station at the exactly same time.

Input

The first line of input contains the number of test cases z ($1 \leqslant z \leqslant 5\,000$). The descriptions of the test cases follow.

The first line of a test case contains one integer n ($1 \leqslant n \leqslant 500\,000$) – the number of trains. The i-th of the next n lines contain space separated departure and travel time of the i-th train. Departure time is given in the form of space separated weekday name, consisting of lowercase English letters, and 24-hour time format `HH:MM:SS` [1]. Travel time is given in the form of x`h`y`m`z`s` ($0 \leqslant x \leqslant 120$, $0 \leqslant y, z \leqslant 59$). Each run takes at least 1 second and at most 120 hours.

The total number of train departure descriptions in all test cases does not exceed 2 000 000.

Output

For each test case, print in the first line one integer d – the smallest number of destinations, so that the trains can run on schedule. In the next d lines print any valid assignment of trains in the following form. In the i-th line print, in a space separated fashion, size of the i-th group followed by indices of trains assigned to that group.

[1] from `00:00:00` to `23:59:59`.

Example

For an example input:	a possible correct output is:
2 4 sunday 14:23:35 47h34m49s wednesday 23:59:59 48h0m0s monday 00:00:00 18h12m34s sunday 09:03:00 15h59m5s 7 tuesday 15:03:10 18h12m34s tuesday 15:03:10 18h12m34s thursday 20:59:58 3h0m1s thursday 20:59:59 3h0m0s friday 16:03:00 15h43m17s friday 16:03:00 15h59m5s saturday 07:26:45 0h57m23s	2 1 3 3 2 4 1 1 7 1 2 3 4 5 6 7

Explanation

In the first test case, first and third train have to run to different destinations (it is a weekly timetable so Sunday train can affect Monday train).

In the second test case, all 7 trains can run to the same destination station.