

Problem 5: Clock Repair

Mr. Horologia's House of Clocks contains various cuckoo clocks that customers have brought in for repair. Since they are in the clock shop, one might rightly assume that these clocks don't quite run as they should. In fact, a fast clock may take 3,500 seconds to advance one hour, instead of 3,600 seconds. A slow clock might take 3,750 seconds.

Every midnight on Sunday morning, Mr. Horologia sets all clocks to exactly 12:00. He has sufficient assistants awake at that hour that all clocks can be set simultaneously. Some time later, possibly that same day but quite possibly several days later, all clocks in the room will cuckoo at precisely the same instant in time. (All clocks initially would chime when they are set at midnight on Sunday, but the initial cuckooing does not count.) What is the first day, hour, minute, and second when all clocks simultaneously go off? The time might be several days in the future; also, midnight on the next Sunday morning might come around again before they ever cuckoo simultaneously. In this latter case, Mr. Horologia will set them all again, and the correct answer would be "Never".

Example

An easy example involves two clocks, one that advances an hour every 3,000 seconds, and a second clock that advances an hour every 4,500 seconds. For simplicity, note that these represent 50 minutes and 75 minutes, respectively. The first clock would reach 1:00 AM after 50 minutes, then 2:00 AM after 100 minutes, and 3:00 AM after 150 minutes. The second clock would reach 1:00 at 75 minutes and 2:00 at 150 minutes. Thus, 150 minutes after midnight, clock number one and clock number two both cuckoo. The correct answer is thus "Sunday at 2:30:00 AM".

A second example might involve four clocks, all of which are extremely fast. They advance at 600, 1200, 1800, and 600 seconds. After 30 minutes, all of them cuckoo except for the second clock. After 60 minutes, all will cuckoo. Thus the answer here is "Sunday at 1:00:00 AM".

Finally, suppose we have four clocks with speeds of 3601, 3559, 3600, and 3700. The answer in this case is "Never" - the next Sunday will occur before all clocks cuckoo.

Input

There will be multiple cases, sequentially numbered starting with 1. The input for each case is a single line that contains integers giving the number of clocks N , followed by N "seconds" measurements. A line containing the integer 0 follows the last case. Because the repair shop is limited, N will never be larger than 10.

Output

For each case, display the case number and either the first date and time when all clocks will cuckoo simultaneously, or the word "Never" as described above. Display a blank line after the output for each case.

Sample Input

```
2 3000 4500
4 600 1200 1800 600
3 3550 3650 3655
2 3525 3625
0
```

Output for the Sample Input

```
Case 1: Sunday at 2:30:00 AM

Case 2: Sunday at 1:00:00 AM

Case 3: Never

Case 4: Friday at 9:58:45 PM
```