Problem H: Rhinoceros Count

Filename: `rhinocount`

Time limit: `1 second`

The Lewa Wildlife Conservancy in Kenya specializes in providing a safe area for Rhinoceros to thrive as Black Rhinoceros in particular are endangered. The 65,000 acre area is home to 186 rhinoceros[1]. To ensure that all 186 are safe, several rangers monitor different sub-areas. Each morning, the rangers go around looking for the rhinoceros. Whenever they find one, they radio in its unique identifier (a number in between 1 and 186, inclusive).

Naturally, it's unlikely that the rangers will see all 186 rhinoceros each day. Instead, if, for four days in a row, if a particular rhinoceros has not been seen, then an extra ranger is assigned to look for it. If a particular rhinoceros hasn't been seen for ten days in a row, then special authorities are notified and a more comprehensive search via airplane with several people is conducted.

The Problem

Given a full log of every ranger's sightings for a list of several days, identify for which rhinoceros need to be assigned an additional ranger because they haven't been seen for four days in a row and which rhinoceros require a full scale search because they haven't been see for ten days in a row. You may assume that the day before the given logs start, all 186 rhinoceros were seen.

The Input

The first line of input contains a single positive integer, $\boldsymbol{c}$ ($1 \leq \boldsymbol{c} \leq 10$), representing the number of input cases.

The first line of each input case contains a single positive integer, $\boldsymbol{n}$ ($4 \leq \boldsymbol{n} \leq 100$), indicating the number of days of ranger logs to analyze for that case. The data for each day, starting with day 1, in chronological order, follow.

The first line of input for each day contains a single positive integer, $\boldsymbol{r}$ ($1 \leq \boldsymbol{r} \leq 30$), representing the number of rangers who make reports on that day. The following $\boldsymbol{r}$ lines will contain the contents of those reports, one line per ranger. Each of these r lines begins with a positive integer, $\boldsymbol{k}$ ($0 \leq \boldsymbol{k} \leq 100$), representing the number of rhinoceros that particular ranger saw. This is followed by $\boldsymbol{k}$ distinct positive integers in the range 1 to 186, inclusive, indicating the numbers of the rhinoceros seen by that ranger on that day.

[1] According to a guide name Abdullah, as of July 23, 2024.

The Output

For each input case, start the output with a single line of the form

```
Case C:
```

Where C is the 1 based case number.

For each day an action is to be taken, output a single line for each action. These lines should be sorted by day, and if multiple actions must be taken on a day, sorted by rhinoceros number.

If a rhinoceros hasn't been seen in the previous 4 days, but has been seen within the last ten days, output a single line with the format

```
Day X: Extra Ranger Y
```

where X is the day the order occurs, Y is the number of the rhinoceros getting an extra assigned ranger.

If a rhinoceros hasn't been seen in the previous 10 days, output a single line with the format

```
Day X: Full Search Y
```

where X is the day the order occurs, Y is the number of the rhinoceros for which a full search is being conducted.

Do not put a blank line between cases.

Note: For each day that an extra ranger or full search occurs, print out a line. (So if there are five days in a row for which a particular rhinoceros isn't seen, but it's seen on the sixth day, then there should be two lines of the form "Day X: Extra Ranger Y" outputted.

Note: Sample Input and Output is too big to be included in the printed copy of the set and is available electronically. However, one sample is provided below where it's assumed that there are 5 total rhinos numbered 1 to 5.

Sample Input **Sample Output**

```
2
11
1
5 1 2 3 4 5
1
4 2 4 1 5
1
4 5 4 1 2
1
4 5 4 1 2
2
3 1 2 5
2 4 2
1
3 2 4 5
1
3 2 4 5
1
3 2 4 5
1
3 2 4 5
1
4 1 2 4 5
1
4 1 2 4 5
5
1
0
1
0
1
0
1
0
```

```
Case 1:
Day 6: Extra Ranger 3
Day 7: Extra Ranger 3
Day 8: Extra Ranger 3
Day 9: Extra Ranger 3
Day 10: Extra Ranger 1
Day 10: Extra Ranger 3
Day 11: Extra Ranger 3
Day 12: Full Search 3
Case 2:
Day 5: Extra Ranger 1
Day 5: Extra Ranger 2
Day 5: Extra Ranger 3
Day 5: Extra Ranger 4
Day 5: Extra Ranger 5
Day 6: Extra Ranger 1
Day 6: Extra Ranger 2
Day 6: Extra Ranger 3
Day 6: Extra Ranger 4
Day 6: Extra Ranger 5
```

Sample Explanation: Rhino 1 isn't found on days 6, 7. 8 and 9, which results in an extra ranger being assigned to find it on day 10 (part of the output), and luckily, with that help, the rhino was found on day 10! Rhino 3 isn't found on days 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. The first day an extra ranger is put on the search is day 6, the first day after the rhino hasn't been seen for four straight days. Finally, on day 12, we must do a full search for that rhino, since it wasn't seen on ten consecutive days, 2 - 11. Notice that we still print this in the output even though the input only has 11 days worth of data. Due to the structure of the output, the only possible days for which actions will be printed are in between day 5 and day n+1.