The Foxens Treasure

There are N ($1 \leq N \leq 4$) Foxen guarding a certain valuable treasure, which you'd love to get your hands on. The problem is, the Foxen certainly aren't about to allow that - at least, not while they're awake.

Fortunately, through careful observation, you've seen that each Fox has a regular sleep cycle. In particular, the ith Fox stays awake for A_i ($1 \leq A_i \leq 23$) hours, then sleeps for S_i ($1 \leq S_i \leq 23$) hours, repeating this pattern indefinitely ($2 \leq A_i+S_i \leq 24$). At the start of your treasure-nabbing attempt, the ith Fox is exactly O_i ($0 \leq O_i < A_i+S_i$) hours into its cycle.

There are T ($1 \leq T \leq 20$) scenarios as described above. For each one, you'd like to determine how soon all of the Foxen will be simultaneously asleep, allowing you to grab their treasure, or if this will simply never happen.

Input

First line: 1 integer, T

For each scenario:

First line: 1 integer, N

Next N lines: 3 integers, A_i, S_i, and O_i, for $i = 1..N$

Output

For each scenario:

1 integer, the minimum number of hours after the start to wait until all of the Foxen are asleep during the same hour. If this will never happen, output the string "Foxen are too powerful" (without quotes) instead.

Example

Input:

```
2
2
2 1 2
2 2 1
3
1 1 0
1 1 0
1 1 1
```

Output:

6

Foxen are too powerful

Explanation of Sample:

In scenario 1, the following table illustrates the Foxen's sleeping cycles (with A representing being awake, S representing sleep, and a bold letter representing the start of a sleep cycle):

Hour	Fox 1	Fox 2
0	S	A
1	**A**	S
2	A	S
3	S	**A**
4	**A**	A
5	A	S
6	S	S

As can be seen, the first hour during which both Foxen are asleep is 6 hours after the start.

In scenario 2, the first 2 Foxen are always awake and asleep at the same times. However, the third Fox's schedule is exactly flipped, which means that it will never be asleep at the same time as the others.