

A competition was just over. It had 3 problems and n players. Each player had an ID number from 1 to n . The final rank was decided by the total score of the 3 problems. The higher the total score was, the higher a player ranked (the smaller the rank number). If two players got the same total score, the one with the smaller ID number got a higher rank. We've known for each problem, how much score each player might get if he didn't solve totally wrong (if solved totally wrong, the player got zero in the problem). However, we don't know whether a player did get score in a problem. For a predicted final rank, you need to judge if the rank is possible.

Input

Input contains several cases. For each case, the first line is an integer n , ($n \leq 16384$) to indicate the number of players, followed by n lines, the i -th of which contains three real numbers a , b , c ($0 \leq a, b, c < 1000$. a , b and c have 2 decimal places at most.) to respectively indicate the score of each problem Player i might get if he didn't solve totally wrong. Another line containing n integers follows to indicate the player ID number in the order from rank 1-st to rank n -th.

The last case is followed by a line containing only a zero.

Output

For each case, if the rank is possible, output the highest possible total score for the player with the lowest rank (calculate to 2 decimal places), otherwise output 'No solution' (quotes for clarity).

Sample Explanation:

Case 1:

Rank	Player ID Number	Problem 1's Score	Problem 2's Score	Problem 3's Score
1	1	100	200	300
2	2	100	200	300
3	3	100	200	300

Case 2:

Rank	Player ID Number	Problem 1's Score	Problem 2's Score	Problem 3's Score
1	3	100	200	300
2	2	0 (wrong)	200	300
3	1	100	0 (wrong)	300

Sample Input

```
3
100 200 300
100 200 300
100 200 300
1 2 3
3
100 200 300
100 200 300
100 200 300
3 2 1
0
```

Sample Output

```
Case 1: 600.00
Case 2: 400.00
```