Shuffle Music

My office is quite far from my home. Every day, I go there by bus. That takes a lot of time. More over, due to traffic jams, the time to reach my office is not fixed. To make the best use of this time, I carry my music player all the time. My music player has a 'shuffle' mode. In that mode, the music player selects a track randomly that has not been played in that session (**before the next Shuffle**), and starts playing. It will continue doing so, until, I have stopped playing, or it has played all tracks in that session.

For the purpose of this problem, you can assume that all tracks are of equal duration. To select randomly, the player keeps list of the tracks already played. That is, if I listen to one track now, from next shuffle session, that track will be marked as played.

In addition, after stopping play, when it shuffles once more, to select a random track, the player always use such strategy that, selecting an old track (**Already listened in previous shuffle session**) is twice more probable than a new track (**Not listened in previous shuffle session**). I go to my office in the morning and return home at night. **Performing 2 shuffles a day, one in the morning, one at night**.

It takes at least p minutes and at most q minutes to reach my office. The time required will always be an integer. You can assume that all travel times are equiprobable (that is, if p = 2 and q = 4, and if P(t) is the probability to reach office in t minutes, then P(2) = P(3) = P(4)). If all tracks are of 1 minute, how many days on average are required to listen to all the songs?

Input

First line of input contains an integer **T(≤10053)**, the number of test cases. Each test case contains 5 integers, **N (1≤N≤75), p1, q1, p2, q2 (0 ≤ p1, q1, p2, q2 ≤ 30, p1 ≤ q1, p2 ≤ q2)**, where **N** is the number of songs, **p1,q1** are the minimum and maximum number of songs listened during day, and **p2,q2** are the minimum and maximum number of songs during night.

Output

For each test case, output the test case, followed by the expected number of days required to listen all songs. If it's not possible to listen to all songs, output "IMPOSSIBLE".

The Judge used is "Ignore FP rounding up to 10^-6".

Example

Input:

```
11
1 0 1 0 1
20 0 26 1 3
10 11 21 0 0
64 0 0 1 23
48 0 0 0 0
42 18 29 6 13
29 19 25 0 0
32 0 0 8 27
```

75 0 1 0 1
75 0 1 0 0
75 0 0 0 1

Output:

Case 1: 1.3333333
Case 2: 3.1143655
Case 3: 1.0000000
Case 4: 39.6303478
Case 5: IMPOSSIBLE
Case 6: 7.1950023
Case 7: 4.7682679
Case 8: 8.1514607
Case 9: 660.4533446
Case 10: 1320.4066892
Case 11: 1320.4066892

Hint

Calculate P [n] [rest] [q] [next]. It is in this session, we have not heard the song in each n songs played, with "rest" number of songs never heard, with q minutes played so far,
 the probability of "next" song been played for the first time. Dp by this $O(N^2q^2)$.
Sum over [0, q] to get P [n] [rest] [qsum] [next], so we can use subtraction to get interval sums.

Now let 0 be day, 1 by night, then we compute dp[0][rest], dp[1] [rest], the expected number of days it takes to finish all the songs if only considering day or night. dp [0] [rest] is [0, rest) of dp [1] [rest] and P [n] [rest] [qsum] [next].
Using $O(rest)$ can be calculated as as X = dp[0][rest], Y = dp[1] [rest], then {X = a Y + c, Y = b X + d}.
Solving this, and done.
In the base case with rest=1, use infinite geometric series to compute overall $O(TN^2 + N^2q^2)$.