

Not that long ago a new railway station was built at Rajshahi, a nice city in Bangladesh. The old one was very small in size. After searching for a while I got a picture of the old station. As you can see the station was mostly open ended in both sides back then. Current railway station is mostly closed in the inner end. In closed end case, the advantage is, we always know where the train will stop and we also know the location of the exit of the train station. So if you want to optimize your time, you can cleverly choose the train bogie and thus exit the whole station at the earliest possible time.

However the situation is not that simple for open ended train stations. The bogie may stop at anywhere at the line. Well, not exactly anywhere, but if we travel for several days we kind of know which bogie usually stops near the exit. Under this circumstances, we would like to compute expected amount of time one requires to exit the station.

For our convenience let us assume the width of a train bogie is negligible and the train line is y -axis. Let the exit of the train station be $(x, 0)$. After traveling several days I know which bogie I should ride. Suppose the length of that bogie is d . We also know that the train randomly stops in the station in such a way that the entire bogie is completely within $(0, y_1)$ and $(0, y_2)$ [$y_1 - y_2 \geq d$]. A bogie has 2 exits, one at the very beginning and another at very end. My sitting location is uniformly random within the bogie. However, when the train stops I look outside through the window and decide which end of the bogie I should exit through to get out of the station at the earliest possible time. If I can walk 1 unit distance per second, what is the expected time for me to get out of the station?



Input

First line of the input will be the number of test cases T ($\leq 1,000$). Hence T test cases follow. Each case consists of a single line with four integers: x, y_1, y_2, d [$-100 \leq y_1, y_2 \leq 100, 0 < d, x \leq 100, y_1 - y_2 \geq d$].

Output

For each test case output: 'Case t : ans ' where t is the case number and ans is the expected amount of time. Precision error of 10^{-4} will be accepted.

Sample Input

```
3
1 1 -1 2
1 1 -1 1
1 10 8 1
```

Sample Output

```
Case 1: 1.914214
Case 2: 1.382597
Case 3: 9.058678
```