

Problem B: Save the Nexus!



You're playing *League of Legends*, and the enemy team is attacking your main building: the Nexus! If they destroy it, they'll win, but fortunately you aren't defenseless.

Your team has all 5 members alive, each of which has their own continuous *damage per second* of D_i . The enemy team also has all 5 members attacking the nexus: each of them has an individual current health of H_i and their *damage per second* of E_i . Your strategy is to attack the enemies. When a given enemy has their health reduced to 0, that enemy will be unable to further attack the nexus. When they are all at 0 health, you've saved the Nexus!

You're confident that the Nexus won't fall to their attack, but you'd like it to take the minimum amount of damage possible. You are able to coordinate your allies together to use the optimal strategy to take out the enemy targets. Any of your members can switch targets at any time, and as many of your allies as you want can attack a single target. The enemy team only attacks the nexus: you can be confident that your teammates will live to tell the tale, and all will be able to output full damage throughout the fight.

What's the minimum amount of damage that the nexus can take, if you coordinate in the optimal fashion?

Input Specification:

The input begins with an integer $T \leq 1000$, the number of test cases. Each test case takes 3 lines, each of which has 5 integers. The first line has 5 positive integers, representing the health of each enemy champion H_i . The second line has 5 positive integers representing the *damage per second* of the enemy champions E_i . Finally, the third line consists of 5 positive integers, the *damage per second* of each of your teammates D_i . Nobody does more than 500 damage per second or has more than 500 health.

Output Specification:

For each test case you should output a single line: the minimum amount of damage that your nexus suffers. As this might not be an integer, you should output it rounded to 5 decimal places.

Sample Input:

```
1
1 2 3 4 5
1 1 1 1 1
1 1 1 1 1
```

Sample Output:

```
7.00000
```