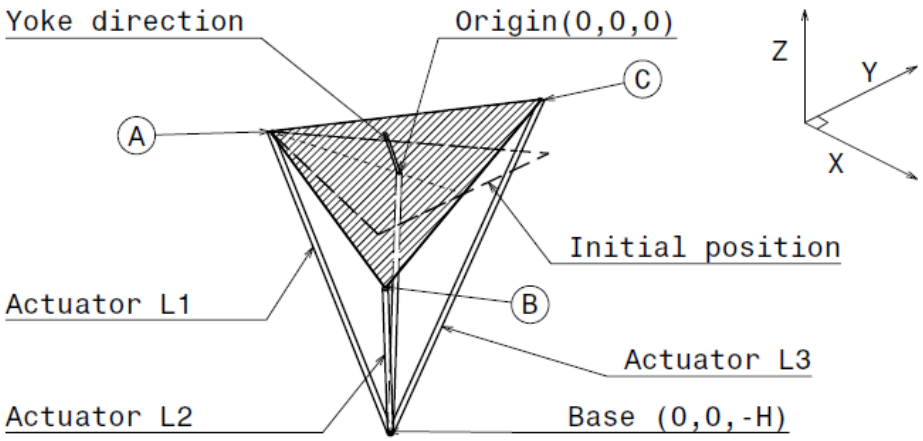
# Problem J. Journey to Jupiter

| | |
|---|---|
| Input file: | `jupiter.in` |
| Output file: | `standard output` |
| Balloon Color: | `Fucia` |

It's the year 10,007 AD, and all the delicious food in the Arab region has gone extinct. Luckily for you, you made once a time travel in the past and you could taste how delicious this food was. An alien rumor has come to you saying that there's still remaining Arab food in Jupiter. Since the Alien rumors, unlike humans', are usually true, you decided to build your own spacecraft to go to Jupiter. In order to test the control system you embedded in your spacecraft, you want to build a flight simulator, which is a device that artificially re-creates spacecraft flight and the environment in which it flies, for astronaut training and other purposes.

In this problem, a flight simulator consists of a 2-dimensional equilateral triangle of side length $L$, with three actuators (mechanical devices, that can be treated as line segments of modifiable lengths) attached to the vertices $A, B, C$ of the triangle, respectively. The other ends of the actuators are all attached to a static base point located at $(0, 0, -H)$. The midpoint of the triangle (located at $(0, 0, 0)$) is attached to the same base using a spherical connection, which means that the triangle can rotate in any direction but its midpoint cannot move.

The yoke (a device used to control the rotations of the triangle) has the direction vector along $\mathbf{N} = (N_x, N_y, N_z)$ which is always normal on the plane of the triangle. By changing the direction of the yoke, the length of each actuator changes and the triangle rotates (the yoke will remain normal on the triangle), while the midpoint remains at $(0, 0, 0)$. Initially, all the actuators have the same length, the triangle lies horizontally in the $XY$ plane (i.e. $z = 0$) and the yoke is pointing along the positive $Z$-axis (i.e. initially $\mathbf{N} = (0, 0, 1)$). The point $A$ is initially located on the negative X-axis ($z_A = y_A = 0, x_A < 0$) and the vertices $A, B, C$ are in an anticlockwise order relative to the yoke direction.




We want you to write a program that makes the simulator rotate given a new yoke direction. Formally, given both of a yoke direction and the location of the point $A$ after rotation, we need you to calculate the corresponding lengths of the actuators $L_1, L_2, L_3$, which are attached to the vertices $A, B, C$ respectively, so that the yoke direction is still normal on the triangle.

## Input

The first line of the input contains a single integer $T$ specifying the number of test cases.

Each test case consists of a single line containing eight integers $N_x$, $N_y$, $N_z$, $A_x$, $A_y$, $A_z$, $L$, and $H$ ($-500 \le A_x, A_y, A_z, N_x, N_y \le 500$, $1 \le N_z, H, L \le 500$), in which $N_x$, $N_y$, and $N_z$ is the yoke direction

after rotation, $A_x$, $A_y$, and $A_z$ is the position of the point $A$ after rotation, $L$ is the side length of the triangle, and $H$ is the point in the base where actuators are attached

It is guaranteed that the following constraints hold for all test cases:

- $A_x, A_y, A_z \neq 0$
- $A_x N_x + A_y N_y + A_z N_z = 0$
- $A_x^2 + A_y^2 + A_z^2 = \frac{L^2}{3}$.

## Output

For each test case, print a single line containing three space-separated decimal numbers (rounded to exactly 6 decimal places), the lengths $L_1, L_2, L_3$ of the actuators connected to the vertices $A, B, C$ corresponding to the given query.

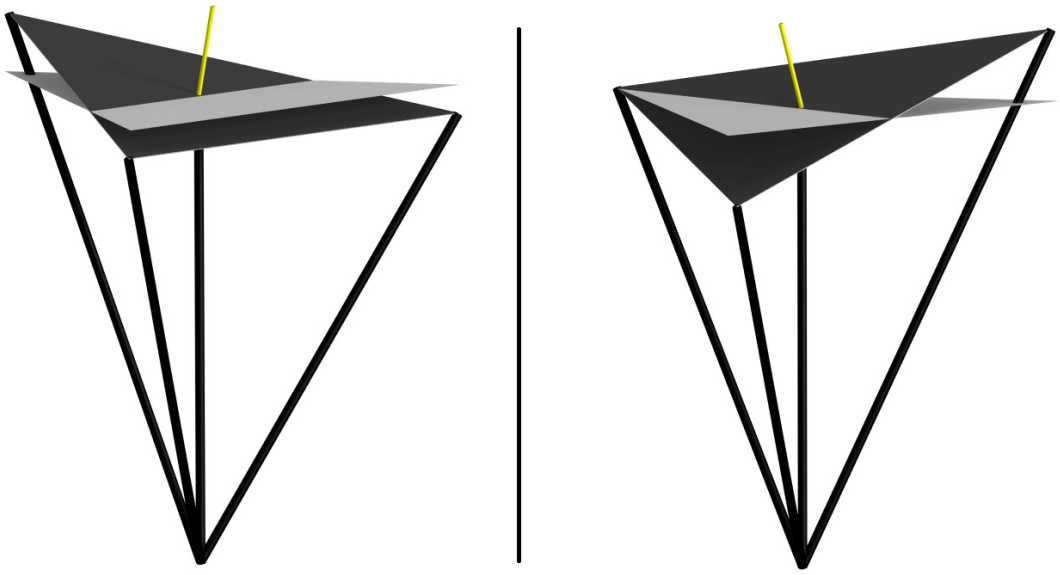
The given input will guarantee that there is exactly one solution. The numbers will be checked with a relative error.

## Example

| jupiter.in | standard output |
|---|---|
| 2<br>13 -18 1 7 5 -1 15 12<br>-12 -17 1 7 -5 -1 15 3 | 13.964240 20.238885 7.237923<br>8.831761 11.473743 6.507936 |

## Note

The figures below show different positions of the flight simulator for different yoke directions :



Rotation in XZ plane

Rotation in YZ plane