Aim

Aim, you fool!

You are a servant aboard a crusading starship just entering a galaxy. A battle ensues between your fleet of four ships and a small armada of natives. Your shields make the vessel you command nearly invulnerable to attack and your weapons can destroy most of your foes in one hit. Unfortunately the Prior in command is less than capable and cannot aim. Thus, when he chooses a target the ship must automatically aim for him. The builders of the ship, however, assumed some competence so there is no such feature and you must create it. Given a target in 3d space with a given relative course and speed and given the speed of the energy missiles fired by your vessel, produce a firing solution, if any, which will cause a shot to strike the enemy vessel.

The first line of the input file contains how many problems to solve.
Each line thereafter will contain:

(x1, y1, z1), [v1, v2, v3], s$

Where x1, y1, z1 are the coordinates of the vessel to be fired upon,
And v1,v2,v3 represent its directional velocity. The last number, variable s, is the speed of your weapon projectile. Assume your ship is at the origin with no velocity, and determine the directional velocity to fire your missile.

For each line of input.txt you need to write to output.txt the direction to fire your missile, or “No Solution” if your missile cannot intercept the enemy.

[v1, v2, v3]$

or

No Solution$

Note that the magnitude of [v1,v2,v3] should be equal to variable s.
The solution vector should satisfy the constraint that it creates an impact in the future (time is positive). All decimals should be rounded to 3 significant places. Use the following rounding code to ensure that your solution is exactly the same as the judge’s:

```
fout.precision(3);
fout.setf(ios::fixed);
fout.setf(ios::showpoint);
```

Example:

input.txt

```
3$
0 100 0 1 0 0 3$
0 100 0 5 0 0 3$
10 100 5 1 -6 .7 3$
```

output.txt

```
1.000 2.828 0.000$
No Solution$
1.812 2.120 1.106$
```