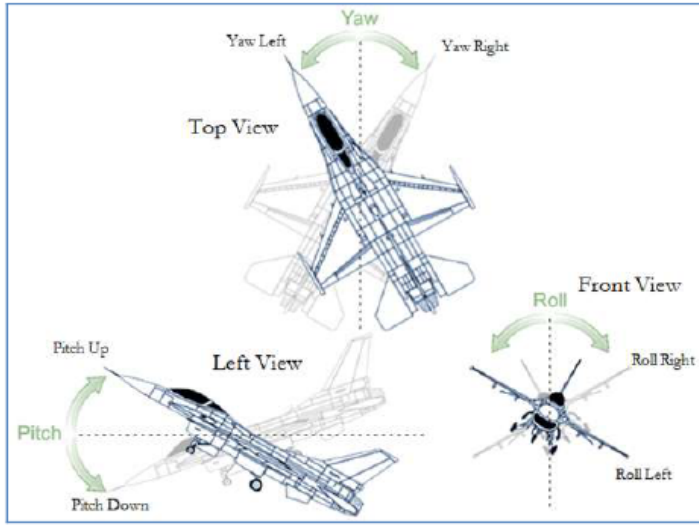


There is a war going on in space. Being a pilot, you need to fly a spaceship in the 3D environment. What makes it more interesting is that there is no predefined “Up” or “Down” direction, due to lack of gravity. One day, you are given a task of observing some object P. You start from a position S. You then visit various positions. Normally your spaceship travels in a straight line. It can stay stationary as well. You finish your journey by reaching the point D. In the whole path you keep observing P.

Now, your superior officer wants to know the following information: “During your travel (i.e. any point on the path from S to D), what was the minimum and maximum distances from the object P to your spaceship?” Very easy task! You just need to look-up the coordinates you have been! Unfortunately, your tracking system got jammed. It could not track the coordinates. But don’t be so disheartened. It did record control commands for the spaceship.



You can rotate your spaceship in three different axes (remember, 3D right?). Refer to that of the movement of a F16 airplane in the figure. You have three kinds of rotations, namely: Pitching, Yawing, and Rolling. These rotations change the directions of the front nose (i.e. moving direction). Additionally, your spaceship can move forward along in the direction of its nose. Of course, argument for this movements and rotations will be given, in **kilometers** (km) and **degrees** respectively. There will four kinds of commands in general:

Command	Remarks
FORWARD < space >> DIST >	$0 < DIST \leq 1000$
PITCH < space >> DEG >	$-180 \leq DEG \leq 180$
YAW < space >> DEG >	$-180 \leq DEG \leq 180$
ROLL < space >> DEG >	$-180 \leq DEG \leq 180$

When viewing your spaceship from outside, 6 rotations can be possible based on the sign of *DEG*.

Left View from outside		Top View from outside		Back View from outside	
PITCH +ve	Pitch up	YAW +ve	Yaw right	ROLL +ve	Roll right
PITCH -ve	Pitch down	YAW -ve	Yaw left	ROLL -ve	Roll left

The 3D space is modeled by coordinates  $(x, y, z)$  each represented in kilometers. So, S and P are given by  $(S_x, S_y, S_z)$  and  $(P_x, P_y, P_z)$  respectively. The spaceship initially is at position S facing the Positive X-axis direction  $(1, 0, 0)$ , and it’s “Up” direction is the Positive Z-axis direction  $(0, 0, 1)$  (Don’t confuse this “Up” direction with “Up” direction of the environment!). Consider this example:  $S = (10, 10, 10)$ . You traveled 30 km from S. Then, you pitched up by 90 degrees, and then went for another 10 km. After that you rolled left for 90 degrees, and went for 30 km.

No.	Command	Spaceship Location	Direction of Moving	Up Direction
	<b>Initial Configuration &gt;&gt;&gt;</b>	(10, 10, 10)	(1, 0, 0)	(0, 0, 1)
1	FORWARD 30	(40, 10, 10)	(1, 0, 0)	(0, 0, 1)
2	PITCH 90	(40, 10, 10)	(0, 0, 1)	(-1, 0, 0)
3	FORWARD 10	(40, 10, 20)	(0, 0, 1)	(-1, 0, 0)
4	ROLL -90	(40, 10, 20)	(0, 0, 1)	(0, 1, 0)
5	FORWARD 30	(40, 10, 50)	(0, 0, 1)	(0, 1, 0)

Given *N* commands from the starting position S sequentially, you need to find the minimum and maximum distances of the object P from your path of travel.

### Input

Input starts with an integer *T* ( $\leq 100$ ), denoting the number of test cases. Each case starts with a line containing seven integers in the following order: *N* ( $0 \leq N \leq 1000$ ),  $S_x, S_y, S_z, P_x, P_y, P_z$ . Each of the next *N* lines contains a command (listed above). The commands are given in sequential order. Magnitude of each of the input values will be less or equal to 1000.

### Output

For each case, print the case number, the minimum and then the maximum distance (both in km). Absolute errors less than  $10^{-6}$  will be tolerated. The minimum distance will not be less than 0.5 km.

### Sample Input

```
1
5 10 10 10 30 10 20
FORWARD 30
PITCH 90
FORWARD 10
ROLL -90
FORWARD 30
```

### Sample Output

```
Case 1: 10.000000 31.622776602
```