

Problem F

Flight Control

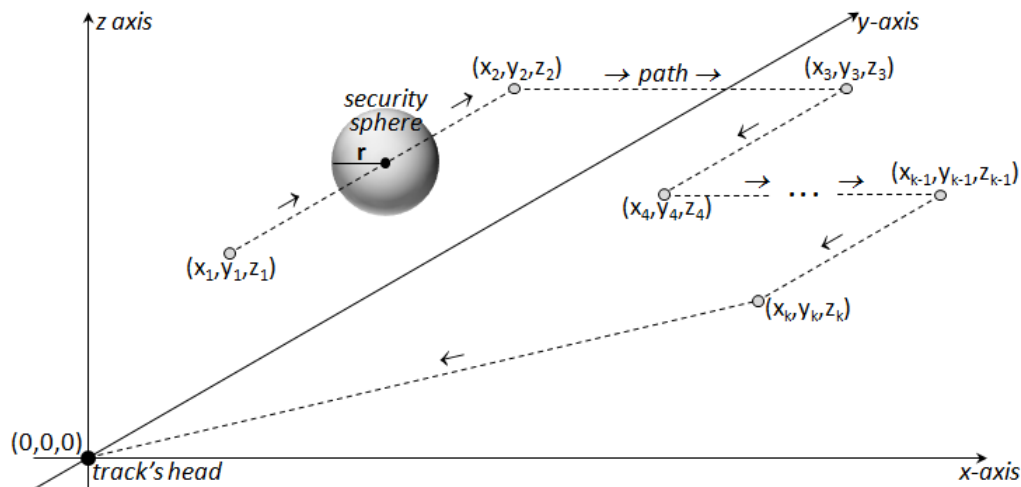
Source file name: `flight.c`, `flight.cpp` or `flight.java`

Air traffic controllers are the people who expedite and maintain a safe and orderly flow of air traffic in the global air traffic control system. The position of the air traffic controller is one that requires highly specialized skills. Because controllers have an incredibly large responsibility while on duty, this profession is, according to Wikipedia, “regarded around the world as one of the most difficult jobs today, and can be notoriously stressful depending on many variables (equipment, configurations, weather, traffic volume, human factors, etc.)”.

An air traffic controller has access to the following information for each aircraft on the screen:

- *size*: a positive integer number r indicating the radius (measured in meters) of a *security sphere* whose center always is the current position of the aircraft;
- *speed*: a positive integer number s indicating the constant speed (measured in meters per second) of the aircraft along its route;
- *route*: a sequence of points $(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_k, y_k, z_k)$ with integer coordinates (measured in meters) in the three-dimensional Cartesian plane, indicating the path followed by the aircraft before it returns to the head of the *track* which is located at the point $(0, 0, 0)$.

Each aircraft begins its journey at the position (x_1, y_1, z_1) and then, it directly flies in a straight line at constant speed from (x_1, y_1, z_1) to (x_2, y_2, z_2) , ..., from $(x_{k-1}, y_{k-1}, z_{k-1})$ to (x_k, y_k, z_k) , and finally from (x_k, y_k, z_k) to $(0, 0, 0)$, where each position is relative to the head of the track. After the aircraft arrives at the head of the track, it disappears from the controller’s screen.



On a day-to-day basis, air traffic controllers deal with conflict detection and resolution. Therefore, for an air traffic controller is very important to have an alarm system to indicate the specific points where it must take corrective actions to prevent accidents.

It is noteworthy that, geometrically speaking, a *conflict warning* occurs when the security spheres of two aircrafts touch. Formally, a *conflict warning* begins when two aircrafts approach at a distance less than or equal to the sum of the radius of their security spheres, is maintained while this condition is satisfied, and ends when their security spheres stop touching (i.e., when the distance between both is greater than the sum of the radius of their security spheres). Distances are measured with an acceptable error threshold $\varepsilon > 0$.

Despite years of effort and the billions of dollars that have been spent on computer software designed to assist air traffic control, success has been largely limited to improving the tools at the disposal of the controllers. However, today you have the chance to improve the impact of computer software in the air traffic control world.

You have been hired by the *International Center of Planning Control* (ICPC) to determine the quality of the traffic routes defined by the *Aircraft Controller Management* (ACM), through the measurement of the number of dangerous situations that should fix the air traffic controller. Your task is to write a program that, given the information of two aircrafts, determines the number of different conflict warnings that would arise if both aircrafts follow the scheduled route starting at the same time and finishing at the track's head.

Input

The first line of the input contains the number of test cases. Each test case specifies the information of the two studied aircrafts, where each aircraft is described as follows:

- The first line contains three integer numbers r , s and k separated by blanks ($1 \leq r \leq 100$, $1 \leq s \leq 1000$, $1 \leq k \leq 100$), where r is the radius of the security sphere (in meters), s is the speed (in meters per second), and k is the number of points that define the route of the aircraft.
- Each one of the next k lines contains three integer numbers x_i , y_i , and z_i separated by blanks ($-10^4 \leq x_i \leq 10^4$, $-10^4 \leq y_i \leq 10^4$, $0 \leq z_i \leq 10^4$), describing the coordinates (in meters) of the i -th point on the route of the aircraft ($1 \leq i \leq k$).

For each route, you may assume that either $x_i \neq x_{i+1}$, $y_i \neq y_{i+1}$ or $z_i \neq z_{i+1}$ for all $1 \leq i < k$, and that either $x_k \neq 0$, $y_k \neq 0$ or $z_k \neq 0$. The acceptable error threshold is $\varepsilon = 10^{-10}$ meters.

The input must be read from the file flight.in.

Output

For each test, print one line informing the number of different conflict warnings.

The output must be written to standard output.

Sample Input	Sample output
7	0
20 300 2	1
10000 1000 5000	2
1000 100 500	1
20 100 3	1
10000 1000 2000	0
1000 100 500	1
100 0 0	
20 300 2	
0 10000 5000	
0 -10000 5000	
20 300 3	
10000 0 5010	
-10000 0 5010	
-8000 1000 3010	
20 300 2	
0 10000 5000	
0 -10000 5000	
20 300 2	
10000 0 5010	
-10000 0 5010	
25 200 2	
3000 6000 3000	
4000 5000 3000	
25 200 2	
3000 6000 3005	
4000 5000 3005	
20 300 2	
5000 4000 3000	
4000 5000 3000	
20 300 2	
3000 6000 3005	
4000 5000 3005	
10 100 1	
-1000 0 0	
10 100 2	
1000 21 0	
-1000 21 0	
10 100 1	
-1000 0 0	
10 100 2	
1000 20 0	
-1000 20 0	