

## 4666 Airport

Martin is the conductor of an airport in Mars. As the Martian scientists invented a new kind of aircraft, Martin is going to have a nightmare.

The new aircraft looks somewhat like a sphere with different radii according to the model. With the powerful engine, the aircraft can accelerate itself to a certain speed in no time, and travel in that speed and direction until stopped. Furthermore, its amazing braking system is able to stop the aircraft immediately.

Unfortunately, due to the global financial crisis, the Martian can't afford the navigation system of the aircraft. That leads to a serious problem: the aircrafts may hit each other while travelling.

Martin's airport is going to replace the old aircraft with the new ones. If any collision happens, Martin will lose his job. Martin has got the speed and radius of every aircraft, together with their travelling plans. He wants to know when the first collision happens, so as to take some actions to avoid the accident. You are the most brilliant programmer in Mars, please help Martin!

### Input

The input consists of multiple test cases.

In each case, the first line consists of an integer  $n$  ( $0 \leq n \leq 1000$ ), which is the number of aircrafts. Each of the next  $n$  lines describes an aircraft. The description for each aircraft consists of 9 integers separated by single space:  $x_i y_i z_i x'_i y'_i z'_i r_i t_i v_i$ .  $(x_i, y_i, z_i)$  and  $(x'_i, y'_i, z'_i)$  indicate the initial position and destination of the center of  $i$ -th aircraft.  $r_i$ ,  $t_i$  and  $v_i$  indicate the radius, starting time and speed of the aircraft. The aircraft will stay where it was when stopped. It is guaranteed that no two aircrafts will touch or overlap with the other initially, and absolute values of all integers will not exceed 10000. Notice that two aircrafts touching each other is considered to be collision.

The last line of input is '-1', which denotes the end of input file.

### Output

For each case, output when the first collision happens with the precision of 0.01 in a single line. If there will never be a collision, just output the word 'Never'.

### Sample Input

```
2
0 0 0 10 0 0 1 0 1
10 2 0 0 2 0 1 0 1
2
1 1 1 3 3 3 2 10 2
-1 -1 -1 -3 -3 -3 1 2 1
4
0 0 0 100 50 20 1 1 2
100 50 20 0 0 0 2 3 4
200 300 150 100 50 20 3 0 1
10 10 10 110 60 30 1 1 2
2
0 0 0 4 5 3 3 0 1
6 7 5 1 2 0 2 5 1
```

```
1
1 2 3 1 2 3 100 10 8
-1
```

### Sample Output

```
5.00
Never
20.76
5.26
Never
```