

Year 2020.

In order to reduce the use of stolen mobile sets in Dhaka city, DMP is planning to place two Advanced Mobile Destroyers (AMDs) on two tall towers of the city. Each AMD has a sphere of action. If a stolen mobile set is found within this sphere (even on the boundary), the AMD erases the mobile set's flash memory, making the set useless.

But there is a bug in the program of the AMD itself: If a stolen set is on the plane passing through the common points of the boundaries of the spheres of two AMDs, no AMD takes any action at all!

Given the position and the radius of the sphere of two AMDs and the position of a stolen set, you are to determine the fate of the set.

## Input

Input consists of several test cases.

Each case consists of case name followed by 11 integers on the same line. The case name contains exactly 5 alphanumeric characters. The first 4 integers are the position  $(x, y, z)$  and the radius  $r$  of the sphere of AMD#1. Next 4 integers are the position  $(x, y, z)$  and the radius  $r$  of the sphere of AMD#2. The last 3 integers are the position  $(x, y, z)$  of the stolen set. Each integer in the input is smaller than 32000 in absolute value. The two spheres of action always intersect each other.

## Output

For each data set, print 4 lines.

The first line gives the case name. The second line and third line indicates whether the stolen set is within the range of AMD#1 & AMD#2 respectively ('y' if within range, 'n' otherwise). The fourth line is 'y' if the stolen set has been made useless, 'n' otherwise.

## Sample Input

```
Dhk01 0 0 0 8 10 0 0 8 5 5 5
Dhk02 0 0 0 8 10 0 0 8 5 0 0
Dhk03 0 0 0 8 10 0 0 8 0 8 0
```

## Sample Output

```
Dhk01
n
n
n
Dhk02
y
y
n
Dhk03
y
n
y
```