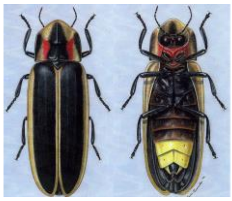

Problem 18: Lampyridae Teleportae

Source filename: `firefly.(cpp|java)`
Input filename: `firefly.in`
Output filename: `firefly.out`

The discovery of a remarkable new insect, the Lampyridae Teleportae, more commonly known as the teleporting firefly, has sparked a no-less-remarkable number of ways to try to catch them. Rather than flying, the Lampyridae Teleportae teleports from spot to spot by warping space-time. When it stops between teleports, it hovers for a moment and flashes its light in search of a mate. So, even though they only come out after dark, it's easy to observe them, but very difficult to catch them. Fortunately for the Association for Catching Lampyridae (ACL), student members of the Association for Cool Machinery (ACM) recently developed the world's first teleporting tennis shoes. The tennis shoes are efficient enough that, when a Lampyridae Teleportae is spotted by its flash, there is always time to teleport once before the firefly itself teleports off to another location, but there is never time to teleport twice in a row before the firefly teleports away. The tennis shoes have a maximum teleport range, however, depending on how well their flux capacitor is constructed, so it's not always possible to catch a Lampyridae Teleportae with just a single teleport. The most efficient catching method is to remain in place until a firefly flashes, and to then teleport in a straight line directly toward it, subject to the limitation of the maximum range of the tennis shoes, in an attempt to get close enough to catch it. If you don't get close enough, you wait for the next flash, teleport towards it again, and repeat, until you either catch it or it's gone.

For this programming problem you will simulate this procedure for catching teleporting fireflies with a few simplifying assumptions: (1) We will be chasing only one firefly at a time. (2) Firefly chasing will take place in two dimensions where all units are considered to be yards. (3) The firefly is "caught" if the chaser can manage to come within one yard of the firefly. (4) The chaser's movement toward a firefly is always in a straight line from his or her current location directly toward the flash; if the range of the chaser's tennis shoes prevents getting close enough to catch the firefly, the chaser will always teleport the maximum range possible (thus, although the chaser always starts at integer coordinates, it is possible and likely that any or all of the chaser's locations after the first teleport will be at non-integer coordinates).

The input will consist of several chase scenarios. For each scenario you will be given the maximum range in yards of the chaser's teleporting tennis shoes, the chaser's starting location, and a list of one or more flash locations for the firefly being chased. For each chase scenario your program will output a single line indicating either the flash location where the firefly was caught, or a message noting that the firefly was never caught.

Input: The first line of a chase scenario contains three numbers, delimited by a single space, in the following order: the maximum range in yards of the chaser's teleporting tennis shoes, the starting x-coordinate of the chaser, and the starting y-coordinate of the chaser. The maximum range will be a positive integer from 1 to 1000. The x and y values for the starting coordinates will be integers from 0 to 1000. The remaining lines of an input scenario contain two integers each, an x-coordinate and a y-coordinate, again delimited by a single space. These are, in order of appearance, the locations where the firefly flashes. All coordinate values range from 0 to 1000. A line specifying a value of -1 for both x and y terminates the list, at which point we consider the firefly to disappear never to be seen again. Note that a firefly might be caught at a flash location prior to end of the list; in this case the rest of the flash locations listed in the input for the current chase scenario should simply be ignored.

The next input scenario begins on the line immediately after the last line of the preceding scenario. An input scenario that specifies 0 (zero) as the maximum range of the chaser will terminate the input.

Output: Every output line will be either: (1) "Firefly N caught at (x,y)", where N is the input scenario number starting with 1, and (x,y) is the last location the firefly flashed before it was caught; or (2) "Firefly N not caught".

Example Input:

```
2 0 0
3 3
4 4
5 5
6 6
7 7
-1 -1
2 0 0
3 3
5 5
7 7
-1 -1
10 50 50
50 62
40 55
30 55
45 45
50 50
55 55
50 50
-1 -1
0 0 0
```

Example Output:

```
Firefly 1 caught at (6,6)
Firefly 2 not caught
Firefly 3 caught at (50,50)
```