Ali Needs a Hug

Filename: hug

Dr. Orooji has spent the last three days at UCF preparing for the High School Programming Contest without food, water, or sleep. Ali needs a hug.

Fortunately, he knows several people at the contest that will gladly hug him if he gets close enough, and they've all gathered in the contest area. Dr. Orooji is still very busy so he will only cross the room once, starting from the west wall and walking only due east. Some people like Ali more than others, and will move farther to hug him than others, so Ali must be careful when deciding where to walk to yield the most of hugs.

The Problem:
Given the locations of Ali's friends in the rectangular room, along with how far each is willing to walk to give Ali a hug, determine the most hugs Ali can get by walking straight across the room from the some point along the west wall directly to the east. Ali is patient, so he will wait for hugs for as long as it takes each person to reach him.

The Input:
There will be multiple data sets. Each data set will begin with a line containing two positive integers, *w* and *h*, separated by a single space, which specify the width and height of the contest area for that data. A room of size 0, 0 indicates the end of the input data and should not be processed.

Following the dimensions of the room is a line containing a single integer, *P*, representing the number of people Ali knows in the contest area. The next *P* lines each contain three non-negative integers, *x*, *y*, and *d*, separated by spaces. *x* and *y* ($0 \leq x \leq w$, $0 \leq y \leq h$) are the x- and y-coordinates of the person in the room (their distance from the west and south walls, respectively). *d* represents the maximum distance at which they will move to hug Ali.

The Output:
For each data set, print a line containing only the following message:

```
Ali can get n hug(s)!
```

Where *n* is the maximum number of hugs Ali can receive.

Sample Input:
```
10 10
3
5 5 5
2 8 2
10 0 5
0 0
```

Sample Output:
```
Ali can get 2 hug(s)!
```