11911 Farther or Closer

In an elementary English class kids are learning the notions of being far or near. To reinforce the kids' sense of these abstract concepts, the teacher introduces his newly invented little class activity called *Farther or Closer* (well it's not really a new invention; we all know the traditional indoor game called *Hotter Colder*:

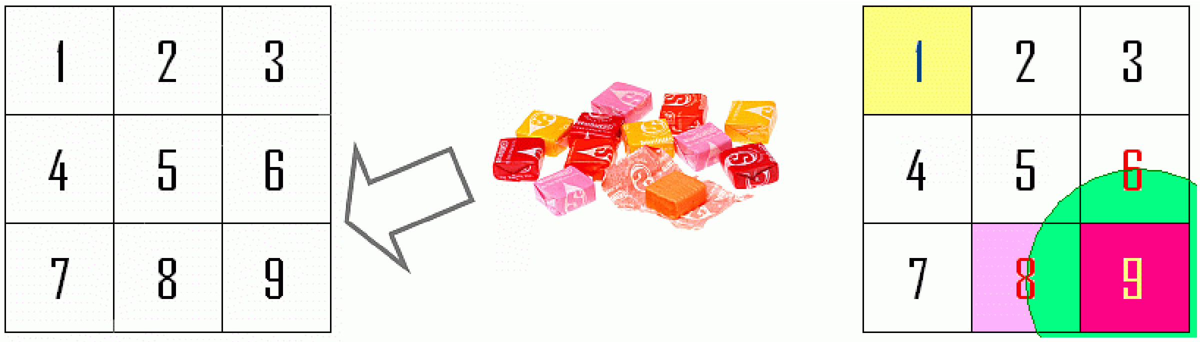

Figure 1: 9 boxes, arranged into a 3-by-3 array   Figure 2: Box 8 is *closer* to 9 than to 1

The teacher has prepared some boxes, arranged into a square as shown in Figure 1 above. As you can see, the numbers are in row-major order. The game roughly goes as follows: the player is to locate the box with a candy hidden inside, with reference to the hints given by other students.

In each play, one student is chosen to be the *guesser*, who first turns around when others put a candy into one of the boxes. All the other boxes are empty. Then, the guesser starts making his guesses, one after another. Here a *guess* is the number written on an available box. If a guess is correct, the students all shout '`YES!!`' loudly and the box is opened. The guesser then gets the candy inside and the game is over. If the guess is incorrect, however, the guesser receives one of the following three replies from the other students:

- `FARTHER` if the correct box is farther away from this guess than from the previous one;
- `CLOSER` if it is nearer to the current guess; or
- `SAME` if the distances are the same.

By *distance* we mean the Euclidean distance between the centres of the boxes. For example, in Figure 2, the distance between boxes 8 and 9 is one unit, and that between 1 and 8 is $\sqrt{5}$ units. Hence, box 8 is closer to 9 than to 1.

For the very first guess the situation is a little bit different. The only two possible replies are '`CLOSER`' (if the chosen box does not contain the candy) or '`YES!!`' (if it does).

Under these rules, every player eventually gets a candy as a reward, and therefore there is no question of finding an *optimal strategy* to *win* the game. Nevertheless, the kids have come up with a clever guessing procedure, which works quite pleasingly in many cases. The method goes like this:

1. Firstly, guess box 1.
2. If incorrect, try the largest numbered available box that possibly contains the candy.
3. If still wrong, guess the smallest numbered box that may contain the candy.
4. Repeat steps 2 and 3 until the candy is found.

To see how this method works in action, let us consider the case of 9 boxes, with the candy in box 8 (see Figure 2 again):

In the first run, the player guesses 1, as stated above. The candy is not in box 1, so everyone replies '`CLOSER`'.

The player then guess 9, the largest numbered remaining box. Since box 8 is closer to 9 than to 1, the students reply '`CLOSER`' again. This leaves only two possibly correct boxes: 6 and 8.

Now the guesser tries the smallest numbered available box, 6. But alas, the candy is not there. This time the guess is farther away from the correct box than before ($\sqrt{2}$ vs. 1).

After three guesses the guesser is sure that the candy is in box 8. The game ends after the box is opened, and now he can finally enjoy his candy!

Now your job is to write a program that imitates the above process.

Input

Each input file contains multiple test cases, and each test case starts on a new line.

Every line of the input contains two integers n and m ($1 \leq m \leq n \leq 100$), which denote the number of boxes and the location of the candy respectively. In all cases n will be a square number.

Output

For each case, give the flow of the corresponding game in the format shown in the sample output below. Note that the numbers should be right aligned to the third column. Consecutive sets of output must be seperated by a blank line.

Sample Input

```
9 8
9 2
```

Sample Output

```
  1? - CLOSER
  9? - CLOSER
  6? - FARTHER
  8? - YES!!

  1? - CLOSER
  9? - FARTHER
  2? - YES!!
```