Problem J. The Good, the Bad and the Ugly

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

This problem was supposed to have a nice long legend about the Wild Wild West, but the author did not manage to write it in time, so use the power of your imagination!

Consider a number line. A player initially stands at the position $x = p$. At the beginning of each round, you can say either "+" or "-". After that, the player changes position according to what you said. More precisely, if you say t and the player stood at position x, then he moves to position $x' = x + d_t$, where d_+ and d_- are two integer constants.

You do not know the exact values p, d_0 and d_1, but you know that the player is either the Good, the Bad or the Ugly (yeah, imagination!):

- The Good player has $p = m$, $d_+ = 2$, $d_- = -1$;
- The Bad player has $p = -m$, $d_+ = 1$, $d_- = -2$;
- The Ugly player has either $p = m$ or $p = -m$ and either $d_+ = 1$ and $d_- = -1$ or $d_+ = -1$ and $d_- = 1$.

As you can see, the starting position of the player depends on some integer constant m ($1 \le m \le 1000$)... unfortunately, you do not know it too.

After each round, the player tells you if he now stands at $x = 0$ or not.

It appears that, by playing several rounds, you can uniquely determine if the player is Good, Bad or Ugly. Do it in no more than $30m$ rounds.

In each test, the values m, p, d_+ and d_- are chosen according to the above rules. They are fixed in advance and don't change during the checking process.

Interaction Protocol

This is an interactive problem.

If you want to play a round, print either "+" or "-" on a separate line. In response, you will get a line containing either 1 if the player arrived at position $x = 0$, or 0 if the player stands somewhere else.

If you are ready to guess the type of the player, print a line containing the character "!", a space and one of the words "good", "bad" or "ugly". After that, your program must terminate.

If after playing $30m$ rounds you do not provide the answer, your solution will get a "Wrong Answer" outcome.

To prevent output buffering, flush the output buffer after each printed line: this can be done by using, for example, `fflush (stdout)` in C or C++, `System.out.flush ()` in Java, `flush (output)` in Pascal, or `sys.stdout.flush ()` in Python.

Example

standard input	standard output
0 1	- - ! good