Red and green

In many computer languages (including e.g, Java) division of positive integers is performed by truncating the "correct" answer, so that for instance 13 divided by 3 produces the result 4 (I just can't bring myself to write $13/3 = 4$.) Given a positive integer n let us say that an integer *k* is a *near factor* of n, if there is some $2 \leq d \leq n$ such that n divided by d produces the result k. For instance, the near factors of 13 are:

$$1, 2, 3, 4, 6.$$

The positive integers are going to be divided into two groups, called green and red, according to the following rules:

- 1 is green.
- A positive integer $n > 1$, n is red if more of its near factors are green than are red. Otherwise, it is green.

For instance:

n	Near factors	Type
1		Green
2	1	Red
3	1	Red
4	1, 2	Green
5	1, 2	Green
6	1, 2, 3	Green
7	1, 2, 3	Green
8	1, 2, 4	Red

Task

Input from `stdin` will consist of a series of lines each of which is (supposed to be) a scenario. A scenario consists of a pair of positive integers, a and b (separated by a space) with $a \leqslant b$. The output for a correctly formatted scenario is: `a b <colours>` where `<colours>` is a string consisting of the characters `R` and `G` representing the types of the integers a through b inclusive. If input is incorrectly formatted in any way then the output for that line should be `Bad input: <in>` where `<in>` is a copy of the input.

You may assume that b will be at most ten million.

(Individual)