Four Sages Around an Oak Tree

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	1024 mebibytes

This is a run-twice problem; both runs are interactive.

Around a wide oak tree stand four sages. Each sage is wearing a red, green, or blue hat on their head. Each sage can see the colors of the hats of both their neighbors but cannot see their own hat color (the hat does not hang over their eyes) or the color of the hat of the sage opposite them (which is blocked by the wide oak tree). Each sage must, without communicating with the others, write down the color of their hat on a piece of paper. Develop a protocol such that at least one of the four sages will definitely guess the color of their hat.

Interaction Protocol

The first line contains two integers t and s, denoting the number of test cases and the run number of the solution ($1 \le t \le 100$; $s \in \{1, 2\}$). Following this are descriptions of t test cases, each on a separate line.

If $s = 1$, the test case consists of four words, c_1, c_2, c_3, and c_4, denoting the colors of the hats of the participants ($c_i \in \{\texttt{red}, \texttt{green}, \texttt{blue}\}$). In response, output a line with a single integer m ($1 \le m \le 4$), denoting the number of the sage who, according to your protocol, will guess the color of their hat (if there are multiple, output any).

If $s = 2$, the test case consists of an integer m and two words, ℓ and r, denoting the number of the sage who must guess the color of their hat, and the colors of their left and right neighbors' hats ($1 \le m \le 4$; $\ell, r \in \{\texttt{red}, \texttt{green}, \texttt{blue}\}$; $\ell = c_{(m+2) \bmod 4+1}$; $r = c_{m \bmod 4+1}$). In response, output a line with a word g that is the color of the hat of the m-th sage. The answer will be accepted if $g = c_m$. The interactor is case-insensitive, so you can print each letter in any case (uppercase or lowercase).

To prevent passing information between test cases, the jury has taken the following measures:

- Both runs are interactive. Thus, you will not receive the next test case until you output the answer to all previous ones. After printing each line, do not forget to flush the output stream (for example, by using `fflush(stdout)` or `cout.flush()` in C++, `sys.stdout.flush()` in Python, or `System.out.flush()` in Java or Kotlin). Otherwise, you will likely receive an **Idleness Limit Exceeded** error.
- During the second run, the jury may rearrange the t test cases in any order, which may not match the order in which they appeared in the first run.
- During the second run, the jury may also add counterfeit test cases that do not correspond to any test case from the first run. For these cases, any answer of "`red`", "`green`", or "`blue`" will be considered correct; they are only needed so that your program cannot easily distinguish a real test case from a counterfeit one and pass information between cases. Thus, t in the second run may exceed t in the first run. The total number of real and counterfeit cases cannot exceed 100.

Examples

standard input	*standard output*
1 1 green red blue red	2
4 2 2 blue green 4 red blue 3 red red 4 red blue	blue BLUE blUE Blue