A. Bear and Prime 100

time limit per test: 1 second
memory limit per test: 256 megabytes

This is an interactive problem. In the output section below you will see the information about flushing the output.

Bear Limak thinks of some hidden number — an integer from interval $[2, 100]$. Your task is to say if the hidden number is prime or composite.

Integer $x > 1$ is called prime if it has exactly two distinct divisors, 1 and x. If integer $x > 1$ is not prime, it's called composite.

You can ask up to 20 queries about divisors of the hidden number. In each query you should print an integer from interval $[2, 100]$. The system will answer "yes" if your integer is a divisor of the hidden number. Otherwise, the answer will be "no".

For example, if the hidden number is 14 then the system will answer "yes" only if you print 2, 7 or 14.

When you are done asking queries, print "prime" or "composite" and terminate your program.

You will get the `Wrong Answer` verdict if you ask more than 20 queries, or if you print an integer not from the range $[2, 100]$. Also, you will get the `Wrong Answer` verdict if the printed answer isn't correct.

You will get the `Idleness Limit Exceeded` verdict if you don't print anything (but you should) or if you forget about flushing the output (more info below).

Input
After each query you should read one string from the input. It will be "yes" if the printed integer is a divisor of the hidden number, and "no" otherwise.

Output
Up to 20 times you can ask a query — print an integer from interval $[2, 100]$ in one line. You have to both print the end-of-line character and `flush` the output. After flushing you should read a response from the input.

In any moment you can print the answer "prime" or "composite" (without the quotes). After that, flush the output and terminate your program.

To flush you can use (just after printing an integer and end-of-line):

- `fflush(stdout)` in C++;
- `System.out.flush()` in Java;
- `stdout.flush()` in Python;
- `flush(output)` in Pascal;
- See the documentation for other languages.

Hacking. To hack someone, as the input you should print the hidden number — one integer from the interval $[2, 100]$. Of course, his/her solution won't be able to read the hidden number from the input.

Examples

input	Copy
yes no yes	

output	Copy
2 80 5 composite	

input	Copy
no yes no no no	

output	Copy
58 59 78 78 2 prime	

Note

The hidden number in the first query is 30. In a table below you can see a better form of the provided example of the communication process.

solution	system
2	
	yes
80	
	no
5	
	yes
composite	

The hidden number is divisible by both 2 and 5. Thus, it must be composite. Note that it isn't necessary to know the exact value of the hidden number. In this test, the hidden number is 30.

solution	system
58	
	no
59	
	yes
78	
	no
78	
	no
2	
	no
prime	

59 is a divisor of the hidden number. In the interval $[2, 100]$ there is only one number with this divisor. The hidden number must be 59, which is prime. Note that the answer is known even after the second query and you could print it then and terminate. Though, it isn't forbidden to ask unnecessary queries (unless you exceed the limit of 20 queries).