Problem ID: fourletters

Writing emails to friends can be tricky for a time traveller. Not only do you have to use the correct email address but the correct date as well. And you have to make sure that you don't send a follow-up email to an earlier date. The *Improved Chronological Private Chat* is a far superior tool for having a conversation. It connects to chat rooms of different eras and ensures that the chronological order of the conversation stays intact even when travelling through time.

After trying to send a message to your newest friend Morgan, you are prompted with the following error: `MESSAGE CONTAINS THE BANNED WORD; MESSAGE REJECTED`. Poking around a bit reveals that the chat room you are connected to has a poorly implemented word filter. It filters out whole messages if they contain a certain blacklisted word as a substring. Poking around a bit more reveals that this forbidden word is a string consisting of four letters, but you do not know the exact string yet.

Certainly there must be a way to find out the banned word while adhering to the size limits of chat room messages and without accidentally spamming your new friend too much.

Interaction Protocol

Your submission will be interacting with a special program called the *grader*. This means that the output of your submission is sent to the grader and the output of the grader is sent to the standard input of your submission. This interaction must follow a specific protocol:

Your submission must send requests of the following two types:

- One line of the form "`?` q", where q is a string consisting of lowercase letters `a`-`z` ($1 \leq |q| \leq 10^4$). The grader will respond `yes` if the banned word is a substring of q, and `no` otherwise.
- One line of the form "`!` w", where w is a string consisting of exactly four lowercase letters, the forbidden word determined by your submission.

After every request you should *flush* the standard output to ensure that the request is sent to the grader. For example, you can use `fflush(stdout)` in C++, `System.out.flush()` in Java, `sys.stdout.flush()` in Python, `hFlush stdout` in Haskell, and `CALL FLUSH()` in Fortran.

Your submission may send at most 100 requests of the first type.

Your submission must send exactly one request of the second type. After sending this request, it must terminate with exit code `0` as usual.

Your submission will be accepted if it follows the protocol above and it guesses the forbidden word correctly. If it sends any invalid request or guesses the forbidden word incorrectly, it will be judged as "Wrong Answer". You may safely assume that there is always a correct solution.

Note that for testing purposes, the forbidden word is not necessarily fixed at the start of the interaction, but may be set by the grader at any point (but always in a way that is consistent with its answers to previous requests).

Sample Input 1

```

yes

no

no

yes

no
```

Sample Output 1

```
? checkmate

? mate

? icpc

? check

? chec

! heck
```

In the example interaction above, the output of the grader is on the left and one possible correct output of a submission is on the right.
The empty lines on both sides only serve to emphasise the chronological order of requests and answers. The grader will never output any empty lines.