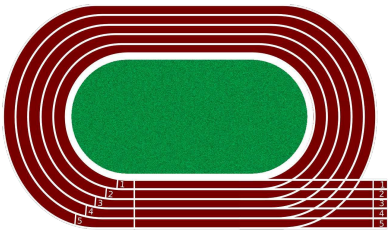
Problem D: Diabolic Doofenshmirtz
Time limit: 1 second



A circular running track. Illustration by Kjrstie, Pixabay

Each year, the fastest platypuses of the world come together for the *Great Crawling Platypus Cup* (GCPC) to crawl a single lap in the platypus arena. Of course, Perry the platypus wants to take part in this great event. Perry's arch enemy Dr. Doofenshmirtz got wind of it and plans to capture Perry right when he crosses the finish line at the GCPC. To carry out his evil plan successfully, he needs to find out how long Perry needs to finish the lap. He knows that Perry takes the preparation for the GCPC very seriously and trains by crawling lap after lap in the arena at a constant speed of $1\ \frac{\text{m}}{\text{s}}$. Perry's training starts this Saturday and Dr. Doofenshmirtz knows that he will run at least 10^{18} seconds. To determine the exact length of one lap, Dr. Doofenshmirtz invented the Measurinator which can measure the exact distance Perry has already crawled in his current lap. Unfortunately, the Measurinator will break after too many measurements, so Dr. Doofenshmirtz has to carefully plan when to use it in order to determine the length of one lap.

Interaction

Your submission can print "`? t`" where t ($0 \leq t < 10^{18}$) is the point in time when you want to use the Measurinator. The time $t = 0$ marks the point in time where Perry starts practicing i.e. where he starts his first lap at the start and finish line. The Measurinator responds with a single integer x which is the current position of Perry in his current lap. Note that the Measurinator will not respond with the number of meters Perry has already crawled in total! Thus, whenever Perry reaches the start and finish line, the Measurinator will respond with 0. Further note that you recently lost your Timetravellinator and therefore, the values of t must be strictly increasing.

If you determined the length x of one lap, print "`! x`". After this, your program should immediately terminate.

All interactions must be ended by a newline. Further note that you additionally need to *flush* the standard output to ensure that the query is sent. For example, you can use:

- `std::cout << std::flush` in C++
- `fflush(stdout)` in C
- `System.out.flush()` in Java
- `sys.stdout.flush()` in Python

Your submission is considered correct if you followed these rules, found the correct length of one lap, and used no more than 42 queries. Note that giving the answer is also counted as a query.

It is guaranteed that one lap has integer length, is at least one meter long, and at most 10^{12} meters long.

A testing tool is provided to help you develop your solution. It can be downloaded from the DOMjudge problems overview page.

Read	Sample Interaction 1	Write
	`? 1`	
`1`		
	`? 2`	
`2`		
	`? 3`	
`3`		
	`? 4`	
`4`		
	`? 100`	
`16`		
	`! 42`	

Read	Sample Interaction 2	Write
	`? 10`	
`10`		
	`? 20`	
`20`		
	`? 30`	
`30`		
	`? 40`	
`40`		
	`? 10000`	
`641`		
	`! 1337`	