

Problem E. Elena Andreeva

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

This is an interactive problem.

You are given an integer n . Your program chooses a number x between 0 and $n - 1$ inclusive. The interactor asks you queries of form: what is the remainder of x modulo y , where y is the parameter of the query.

The interactor isn't stupid and won't ask you a query, such that the answer to it is known beforehand (i.e. it can be deduced from answers to previous queries and the fact that the hidden number is between 0 and $n - 1$). For example, if the interactor asks you the remainder of x modulo 10 and you answer 6 the interactor won't query the remainder of x modulo 2 because it can be deduced that it is 0.

You want the interactor to know the number you've chosen in as few queries as possible. Obviously you can answer adaptively and change the chosen number on the fly, but it must remain consistent with previous answers.

At the start of interaction you choose an integer k and implicitly make a claim: $\lceil \frac{n}{k} \rceil$ The interactor will know the chosen number in at most k queries. You should choose the smallest k , such that you can always make that happen. Precisely, let K denote the smallest possible such k .

If you choose $k > K$ the interactor will tell you that. You can either accept this fact or choose to reverse the roles. You should do it only if you believe that your solution can't be wrong (hint: it is) and there is a problem with authors solution or tests. The interactor will choose the number and you will try to guess it as slowly as possible. If you make k valid queries you will receive Check Failed and prove your point.

If $k \leq K$ the guessing starts. The interactor will make queries until it knows the number. If the interactor is able to make $k + 1$ queries you didn't fulfill your claim and receive WA. Otherwise you pass the test case. Note that if you choose $k < K$ the interactor will always be able to make at least $k + 1$ queries by asking queries optimally (however there is no guarantee that the interactor will do so, refer to the third sample).

Interaction Protocol

You should read an integer n ($2 \leq n \leq 10^5$).

After that you should make your claim. Print an integer k ($1 \leq k \leq 10^5$). There are two possibilities:

1. If $k \leq K$ the interactor prints 1. You choose a number x and the interactor starts guessing.
You should repeatedly read an integer y ($2 \leq y \leq 10^9$) and print $x \bmod y$ until you encounter $y = -1$ or $y = 0$ or the interactor makes the $k + 1$ -th query. $y = -1$ means your answers are inconsistent. $y = 0$ means that interactor knows the number and you've passed the test. $k + 1$ -th query means that you didn't fulfill your claim. Your solution should terminate in all 3 cases.
2. If $k > K$ the interactor prints 0. Repeatedly print an integer y ($2 \leq y \leq 10^9$) and read an integer r which is equal to $x \bmod y$. Your queries should be valid (you don't know the answer beforehand). If some query is not valid the interactor will print -1 instead of the remainder and your solution should terminate. When you know the hidden number or you simply accept the fact that your solution is wrong print -1 and terminate instead. If you make k valid queries (this is highly unlikely) you will receive Check Failed or something along the lines.

Note that -1 and 0 are signaling values and they are out of bounds $[2, 10^9]$ used for ordinary value of y .

Examples

standard input	standard output
3 1 15 0	1 0
6 1 2 5 0	2 1 0
100 1 80 0	1 76

Note

In the first sample the process goes as follows.

- The solution reads n . It is equal to 3.
- It prints $k = 1$.
- It can be shown that $K = 1$.
- The interactor prints 1 and starts guessing.
- It makes a query with $y = 15$.
- The solutions answers 0.
- Interactor knows that the hidden number is 0 and prints -1.

In the second sample a similar process happens, except for $k = K = 2$. Therefore the interactor makes two queries instead of one. The hidden number is 5.

In the third example $K > 1$, but the solution prints $k = 1$. However, the interactor makes a query, such that the hidden number (76) is known after the solution answers it and the solution still passes the test. If the interactor asked a query with $y = 10$ instead the solution would've failed test because there will always remain multiple possibilities for the hidden number after the query is answered.

The actual interactor might ask different queries than ones in the samples. Therefore there is no easy way to pass the samples with some placeholder solution