Swiss Stig (Part 1)

Background

The car show *Toxic Gas* has just commissioned the Swiss Stig to benchmark a new model, and is asking them to rate the car using a scale from `1` (terrible) to `$maxRating` (perfect).

The Swiss Stig, being a Stig, could not speak but only react in ✔ and ❌. Welp, it's gonna be hard to figure out his score.

To make things worse, the show's interviewer is an absolute pillock. The interviewing process simply repeats the following process for `$Q` times.

```
Interviewer: Is the score at least upper than ${lower}, but not upper than ${upper
Stig: ✔ (or ❌ whenever appropriate)
```

The list of question is predetermined ahead of time. If we cannot determine what's exactly the car's rating, we shall act prudent and choose the lowest possible score.

As an analyst, before the show goes live, you want to convince the interviewer to change his plan by showing how (in-)accurate this methodology is, assuming the underlying rating from Stig is an integer uniformly lying between `[1, maxRating]`.

Example Input

```
[{
   "questions": [{
      "lower": 2,
      "upper": 3,
   }],
   "maxRating": 5,
}]
```

`Example Output`

```
[{
   "p": 2,
   "q": 5,
}]
```

Explanation

The output means the accuracy is 2 over 5, that is, `p/q = 2/5`.

When `rating = 1`, Stig replies ❌ and interviewer reports 1 for prudence, that is accurate.

When `rating = 2`, Stig replies ✔ and interviewer reports 2 for prudence, that is accurate.

When `rating = 3`, Stig replies ✔ and interviewer reports 2 for prudence, that is inaccurate.

When `rating = 4`, Stig replies ❌ and interviewer reports 1 for prudence, that is inaccurate.

When `rating = 5`, Stig replies ❌ and interviewer reports 1 for prudence, that is inaccurate.

In summary, the interview is accurate 2 out of 5 times, hence the output.

What we know so far ...

Input

```
Question is {
   "lower" is Int,
   "upper" is Int
}
Interview = {
   "questions" is List<Question>,
   "maxRating" is Int
   // we will add more fields in full task,
}

Input is List<Interview>
```

Output

```
Accuracy is {
   "p" is Int,
   "q" is Int,
}

Output is List<Accuracy>
```

Furthermore, Accuracy must be irreducible, i.e. `gcd(p, q) = 1`

Warmup Subtask

```
Interview.size() <= 1'000
maxRating <= 1'000
Input.size() = 5
```

Scoring

Subtask total: 400

For each correct `Accuracy` you are rewarded 80 points.

Endpoint

`/stig/warmup`

Let's continue the story ...

Listening to your opinion, the interviewer have made a slight change to his methodology. He believes the inaccuracy is due to he never change his questions after each response of the Stig (Yay!), so he decided to adjust his questions described as follows, using his favourite lucky number `$x` .

After asking the first `$i` questions, sample `accuracy = p/q` if the interview only had first `$i-1` questions (accuracy without knowing actual response)

```
f(value) -> (value + $p * $x - 1 ) Modulo $maxRating + 1
lower <- f(lower)
upper <- f(upper)

swap lower and upper whenever appropriate
```

VISIBLE analyst sigh What a weird and meaningless method.

We shall show the interviewer once again how inaccurate this is, shouldn't we?

Full subtask

Input

```
Question is {
   "lower" is Int,
   "upper" is Int
}

Interview is {
   "questions" is List<Question>,
   "maxRating" is Int,
   "lucky" is Int,
}
```

Constraints

```
Interview.size() <= 10^5
maxRating <= 10^9
lucky <= 10^9
Input.size() = 5
```

Output and Scoring

Same as warmup subtask

Endpoint

`/stig/full`

Time Limit

10 seconds for each input / 5 interviews.