



Programming Languages II

Unless otherwise stated, exercises should be submitted in electronic form, via the collaborative learning environment moodle.softlab.ntua.gr. Deadlines will be strict. You are allowed at most one late exercise.

Exercise 3 Lose twice and you're out

Deadline: 15/12/2019

Write two Haskell programs that solve the following task. The first program must use purely functional data structures to solve the task, whereas the second may use impure data structures, such as arrays in some appropriate monad. Submit your solutions (separately) to the automated grading system available at grader.softlab.ntua.gr.

Task description. In an online pool tournament players play against each other in pairs, in the order dictated by the system. For each game, the winner earns a number of points and such points are added; in the contrary, the loser earns no points. To make the tournament more interesting, the organizers decided on the following set of rules:

- The points awarded are doubled for every successive win. The first win earns 1 point, the second successive win earns 2 points, the third successive win earns 4 points, and so on.
- In case of a loss, the successive win streak resets. A subsequent win earns again 1 point.
- If a player loses twice in a row, they are eliminated from the tournament.

Let $A \leq B$ be two natural numbers. Find out in how many different ways a player can earn k points, where $A \leq k \leq B$ points, before being eliminated.

For example, consider $A = 4$ and $B = 5$. A player can earn 4 or 5 points in the following fourteen (14) ways, where “ k ” denotes a win that gives k points and “ \times ” denotes a loss:

1 × 1 × 1 × 1 × ×	1 × 1 × 1 × 1 × 1 × ×
× 1 × 1 × 1 × 1 × ×	× 1 × 1 × 1 × 1 × 1 × ×
1 × 1 2 × ×	1 × 1 × 1 2 × ×
× 1 × 1 2 × ×	× 1 × 1 × 1 2 × ×
1 2 × 1 × ×	1 × 1 2 × 1 × ×
× 1 2 × 1 × ×	× 1 × 1 2 × 1 × ×
	1 2 × 1 × 1 × ×
	× 1 2 × 1 × 1 × ×

Input and output. Your program will read from the standard input (stdin) and print the results to the standard output (stdout).

The first line of the input will contain two space-separated integer numbers N and M . The number N is the number of queries that follow. Because the number of ways in which a player can earn points can be very large, for each query you have to print the result modulo M . Each of the following N lines contains two space-separated integer numbers A and B . They represent a query that you have to answer.

The output must consist of exactly N lines, each of which must contain exactly one integer number: your answer to the corresponding query.

Example #1

Input	Output
4 2019	12
2 4	4
3 3	14
4 5	936
17 42	

Example #2

Input	Output
3 1000000007	197
0 10	376826466
1742 4217	969790604
500000 999999	

Explanation. The 3rd query of the 1st example ($A = 4, B = 5$) was explained above. For the 4th query of the 1st example ($A = 17, B = 42$), there are 126.321.690 ways in which a player can earn between 17 and 42 points. As $M = 2019$, the answer is 936 (because $126.321.690 = 62.566 \times 2019 + 936$).

Constraints.

- $1 \leq N \leq 100.000$
- $0 \leq A \leq B \leq 1.000.000$
- Time limit: 3 sec. (pure) and 1 sec. (impure).
- Memory limit: 64 MB.

To obtain a perfect score, your solution must be efficient. Notice that, in Haskell reading the input efficiently, using efficient data structures and avoiding laziness may prove to be difficult tasks.