```
1   // Eugene Ngo
2   // 11/28/2022
3   // EE 271
4   // Lab 5
5
6   // LFSR module takes in 1-bit clk and 1-bit reset as inputs and returns 10-bit Q. This
    module is used to
7   // simulate computer button presses via a random number generator.
8
9   module LFSR(clk, reset, Q);
10     output logic [9:0] Q;
11     input logic clk, reset;
12
13     logic xnor_out;
14
15     assign xnor_out = (Q[0] ~^ Q[3]);
16
17     always_ff @(posedge clk) begin
18        if(reset) Q <= 10'b0000000000;
19
20        else Q <= {xnor_out, Q[9:1]};
21     end
22  endmodule
23
24  // LFSR_testbench tests all expected, unexpected, and edgecase behaviors
25  module LFSR_testbench();
26     logic [10:1] Q;
27     logic clk, reset;
28     logic xnor_out;
29
30     LFSR dut(.clk, .reset, .Q);
31
32     parameter CLOCK_PERIOD = 100;
33     initial begin
34        clk <= 0;
35        forever #(CLOCK_PERIOD / 2)
36        clk <= ~clk;
37     end
38
39     initial begin
40        reset <= 1;                    @(posedge clk);
41                                       @(posedge clk);
42        reset <= 0;                    @(posedge clk);
43                                       @(posedge clk);
44                                       @(posedge clk);
45                                       @(posedge clk);
46                                       @(posedge clk);
47                                       @(posedge clk);
48                                       @(posedge clk);
49                                       @(posedge clk);
50                                       @(posedge clk);
51                                       @(posedge clk);
52                                       @(posedge clk);
53                                       @(posedge clk);
54                                       @(posedge clk);
55        $stop;
56     end
57  endmodule
58
59
```