```
 1   module lfsr_10(clk, reset, out);
 2      input  logic clk, reset;
 3      output logic [9:0] out;
 4
 5      logic [9:0] ps;
 6
 7      always_ff @(posedge clk) begin
 8         if (reset)
 9            ps = 9'b0;
10         else begin
11            ps[0] <= ~(ps[9]^ps[6]);
12            ps[1] <= ps[0];
13            ps[2] <= ps[1];
14            ps[3] <= ps[2];
15            ps[4] <= ps[3];
16            ps[5] <= ps[4];
17            ps[6] <= ps[5];
18            ps[7] <= ps[6];
19            ps[8] <= ps[7];
20            ps[9] <= ps[8];
21            out   <= ps;
22         end
23      end
24   endmodule
25
26   module lfsr_10_testbench();
27      logic clk, reset;
28      logic [9:0] out;
29
30       lfsr_10 dut(.clk, .reset, .out);
31
32       parameter CLOCK_PERIOD = 100;
33
34      initial begin
35         clk <= 0;
36         forever #(CLOCK_PERIOD/2) clk <= ~clk;
37      end
38
39      integer i;
40
41      initial begin                @(posedge clk);
42         reset <= 1;               @(posedge clk);
43         reset <= 0;               @(posedge clk);
44         for (i = 0; i < 2048; i++) begin
45            @(posedge clk);
46         end
47         $stop;
48      end
49   endmodule
50
```