```
1   // Eugene Ngo
2   // 11/18/2022
3   // EE 271
4   // Lab 4
5
6   // centerLight module takes in clk, reset, L, R, NL, NR as inputs and they are all 1-Bit
    and returns
7   // lightOn which is also 1-Bit. This module determines the next state based on the inputs
    and moves the light
8   // to the left if key 3 is pressed and moves the light to the right if key 0 is pressed.
    When the game is reset
9   // this module recenters the light for the next game.
10
11  module centerLight (lightOn, clk, reset, L, R, NL, NR);
12
13     output logic lightOn;
14     input logic clk, reset;
15     input logic L, R, NL, NR;
16
17     enum {on, off} ps, ns;
18
19     always_comb begin
20        case(ps)
21           on:    if(~R & L | R & ~L) ns = off;
22                  else ns = on;
23
24           off:   if(NR & L & ~R | NL & R & ~L) ns = on;
25                  else ns = off;
26
27        endcase
28     end
29     always_comb begin
30        case(ps)
31           on: lightOn = 1;
32
33           off: lightOn = 0;
34        endcase
35     end
36
37     always_ff @(posedge clk) begin
38        if(reset)
39           ps <= on;
40        else
41           ps <= ns;
42     end
43
44  endmodule
45
46  // centerLight_testbench tests all expected, unexpected, and edgecase behaviors of the
    lights.
47  module centerLight_testbench();
48     logic L, R, NL, NR;
49     logic lightOn;
50     logic clk, reset;
51
52     centerLight dut (.lightOn, .clk, .reset, .L, .R, .NL, .NR);
53
54     //Set up the clock
55     parameter CLOCK_PERIOD = 100;
56     initial begin
57        clk <= 0;
58        forever #(CLOCK_PERIOD / 2)
59        clk <= ~clk;
60     end
61
62     //Set up the inputs to the design. Each line is a clock cycle
63     initial begin
64                                          @(posedge clk)
65        reset <= 1;                       @(posedge clk)
66                                          @(posedge clk)
67        reset <= 0;                       @(posedge clk)
68                                          @(posedge clk)
69        L <= 1; R <= 0; NL <= 0; NR <= 1; @(posedge clk)
```

```
                                                @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
                        NL <= 1; NR <= 0;       @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
        L <= 0; R <= 1;                         @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
                        NL <= 0; NR <= 1;       @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
        L <= 1;         NL <= 1; NR <= 0;       @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
        reset <= 1;                             @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
        L <= 0; R <= 1; NL <= 1; NR <= 0;       @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
        L <= 1; R <= 0; NL <= 0; NR <= 0;       @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
        L <= 0; R <= 1; NL <= 0; NR <= 0;       @(posedge clk)
                                                @(posedge clk)
                                                @(posedge clk)
        $stop;
    end
endmodule
```