```
module winner (clk, reset, LED9, LED1, L, R, user);
    input logic clk, reset;
    input logic LED9, LED1, L, R;
    output logic [6:0] user;

    enum {off, P1, P2} ps, ns;

    always_comb begin
        case(ps)
            off:   if(LED9 & L & ~R) ns = P1;

                     else if(LED1 & R & ~L) ns = P2;

                     else ns = off;

            P1: ns = P1;

            P2: ns = P2;

        endcase

        if(ns == P1) user = 7'b1111001;
        else if (ns == P2) user = 7'b0100100;
        else user = 7'b1111111;

    end

    always_ff @(posedge clk) begin
        if(reset)
            ps <= off;
        else
            ps <= ns;
    end

endmodule

module winner_testbench();
    logic clk, reset, LED9, LED1, L, R, user;

    winner dut (.clk, .reset, .LED9, .LED1, .L, .R, .user);

    parameter CLOCK_PERIOD = 100;
    initial begin
        clk <= 0;
        forever #(CLOCK_PERIOD / 2)
        clk <= ~clk;
    end

    initial begin
        reset <= 1; @(posedge clk);
        reset <= 0; @(posedge clk);
        LED9 <= 0; LED1 <= 0; L <= 0; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
                                    L <= 1; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
                        LED1 <= 1; L <= 0; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
                                    L <= 1; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
        LED9 <= 1; LED1 <= 0; L <= 0; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
                                    L <= 1; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
                        LED1 <= 1; L <= 0; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
                                    L <= 1; R <= 0; repeat(4) @(posedge clk);
                                              R <= 1; repeat(4) @(posedge clk);
        $stop;
    end
```

```
70    endmodule
```