Problem 3: [6 points] Drill problem
Filename: hw5prob3.sv
AndrewID: jtbell

```
module hw5prob3
  (input logic A, clock, reset_L,
   output logic B, C);

    enum logic [2:0] {state1=3'b010, state2=3'b100,state3= 3'b011}
      currState,nextState;

    always_comb begin
      case(currState)
        state1: begin
          if(A == 0) begin
            nextState = state2;
          end
          if(A == 1) begin
            nextState = state3;
          end
        end
        state2: begin
          if(A == 0) begin
            nextState = state3;
          end
          if(A==1) begin
            nextState = state2;
          end
        end
        state3: begin
          if (A == 0) begin
            nextState = state1;
          end
          if (A == 1) begin
            nextState = state3;
          end
        end
      endcase
    end

    always_comb begin
      case(currState)
        state1: begin
          B = 0;
          C = 1;
        end
        state2: begin
          B = 1;
          C = 0;
        end
        state3: begin
          B = 0;
          C = 1;
        end
      endcase
    end

  always_ff @(posedge clock, negedge reset_L)
    if (~reset_L)
      currState <= state1; // or whatever the reset state is
    else
      currState <= nextState;

endmodule

module hw5prob3_tb();
  logic A, clock, reset_L;
  logic B, C;

  hw5prob3 DUT(.*);

  initial begin
    clock = 0;
    reset_L = 0;
    reset_L <= 1;

    forever #5 clock = ~clock;
  end

  initial begin
    $monitor($time,, "state=%s, A=%b, B=%b,C=%b,reset_L=%b",DUT.currState.name,
      A, B, C, reset_L);
  end

  initial begin
    A <= 1'b0;
    reset_L <= 1'b1;
    @(posedge clock);
    A <= 1'b0;
    @(posedge clock);
    A <= 1'b0;
    @(posedge clock);
    A <= 1'b1;
    @(posedge clock);
    A <= 1'b0;
    @(posedge clock);
    A <= 1'b1;
    @(posedge clock);
    A <= 1'b1;
    @(posedge clock);
    reset_L <= 1'b0;
    @(posedge clock);
  #100 $finish;
  end

endmodule
```