Exercise 1 – Seg7 Test Bench

Write a test bench for the provided seg7 module from Lecture 3. Be thorough, including all 16 input combinations!

```
module seg7 (bcd, leds);
    input logic [3:0] bcd;
    output logic [6:0] leds;

    always_comb
        case (bcd)
            //          Light: 6543210
            4'b0000: leds = 7'b0111111; // 0
            ... // implementation
            4'b1001: leds = 7'b1101111; // 9
            default: leds = 7'bX;
        endcase
endmodule // seg7

module seg7_tb ();
    logic [3:0] bcd;
    logic [6:0] leds;

    // TODO: Test the module!









endmodule // seg7_tb
```

Exercise 2 – Guessing Game Test Bench

Write a test bench for the Guessing Game module from Section 2. Be sure to think about how many input combinations there are!

```
// Game to check user's 3-bit input guess against a hard-coded
    secret #
// - SW[2:0] is the guess, KEY[0] is check
// - LEDR[0] is <, LEDR[1] is ==, LEDR[2] is >
module guessing_game (
    output logic [9:0] LEDR,
    input logic [3:0] KEY,
    input logic [9:0] SW
);
    ... // implementation
endmodule // guessing_game

module guessing_game_tb ();
    logic [9:0] LEDR;
    logic [3:0] KEY;
    logic [9:0] SW;

    // TODO: Test the module!









endmodule // guessing_game_tb
```