VERILOG CODING TEST

1. Write a Verilog module that blinks an LED at 1 Hz using a 50 MHz clock. The LED must toggle every 0.5 seconds. Inputs are clk and rst, output is led.
2. Implement a 4-bit shift register that shifts a pattern across 4 LEDs.
   Requirements:
   Input: clk, rst, shift_in
   Output: led[3:0]
   Shift every 0.5 seconds
3. Ripple carry adder using genvr
4. Design configurable full adder using parameter construct (Test it for 8 bit and 16 bit full adder). Use $Strobe for Displaying result on transcript
5. Generate a 100 Hz Clock from a 50 MHz Clock in Verilog.
6. Design 8:3 priority encoder using casez statements.
7. Write a Verilog program with the following specifications.
    1. It should have a memory of size 8x14.
    2. Write a task/function to fill all elements in the memory.
    3. Write a method to add the values stored inside the memory.
    4. 4. Write a method to find the largest number in the memory.
8. 1 22 333 4444 55555 4444 333 22 1 display this sequence in diamond shape
9. Design a finite state machine that has an input x and output y. The output should be asserted whenever x = 1 or x = 0 for three consecutive clock pulses. In other words, the FSM should detect the sequences 111 or 000. Overlapping sequences are allowed, so a sequence of four or five 0s or 1s should also output 1.
10. Write Verilog code for sequence detector 10110 in Moore - overlapping
11. Write a verilog code to generate random numbers between -100 to 100.