```
1   // Eugene Ngo
2   // 5/3/23
3   // EE 469
4   // Lab 3
5
6   /* testbench is a simulation module which simply instantiates the processor system and runs
    50 cycles
7   ** of instructions before terminating. At termination, specific register file values are
    checked to
8   ** verify the processors' ability to execute the implemented instructions.
9   */
10  `timescale 1ns/10ps
11  module testbench();
12
13      // system signals
14      logic clk, rst;
15
16      // generate clock with 100ps clk period
17      initial begin
18          clk = '1;
19          forever #50 clk = ~clk;
20      end
21
22      // processor instantion. Within is the processor as well as imem and dmem
23      top cpu (.clk(clk), .rst(rst));
24
25      initial begin
26          // start with a basic reset
27          rst = 1; @(posedge clk);
28          rst <= 0; @(posedge clk);
29
30          // repeat for 50 cycles. Not all 50 are necessary, however a loop at the end of the
    program will keep anything weird from happening
31          repeat(50) @(posedge clk);
32
33          // basic checking to ensure the right final answer is achieved. These DO NOT prove
    your system works. A more careful look at your
34          // simulation and code will be made.
35
36          // task 1:
37          assert(cpu.processor.u_reg_file.memory[8] == 32'd11) $display("Task 1 Passed");
38          else                                                 $display("Task 1 Failed");
39
40          // task 2:
41          //assert(cpu.processor.u_reg_file.memory[8] == 32'd1)  $display("Task 2 Passed");
42          //else                                                 $display("Task 2 Failed");
43
44          $stop;
45      end
46
47  endmodule
```