1. (15 points) (a) (4 points) Create a VHDL entity named `mux_16_to_1` that represents a 16 to 1 multiplexer. (b) (11 points) Create a VHDL architecture representing a structural model of the 16 to 1 mux using as many `mux_4_to_1` muxes as are needed. You do not need to write an entity or an architecture for `mux_4_to_1`. You may also assume that a component has already been declared and that no configuration statement is required.
2. (1 point) The synthesizable subset of VHDL is standard. (True/False) _____

3. (20 points). (a) (12 points) Write a VHDL function that will take two integer vectors, A and B, and find the dot product \( C = \sum ai \times bi \). The function call should be of the form DOT(A,B), where A and B are integer vector signals. Use attributes inside the function to determine the length and range of the vectors. Make no assumptions about the high and low values of the ranges. For example,
   \( A(3 \text{ downto } 1) = (1,2,3), B(3 \text{ downto } 1) = (4,5,6), C = 3*6 + 2*5 + 1*4 = 32 \).
   \( A(0 \text{ to } 4) = (1,3,5,7,9), B(9 \text{ downto } 5) = (2,4,6,8,10) = 1*2 + 3*4 + 5*6 + 7*8 + 9*10 = 190 \)

Output a warning if the ranges are not the same.

(b)(8 points) Show an architecture that includes two calls to the function with the following properties. 1 - returns a value, 2 – triggers a warning message.
4. (1 point) All processes are executed at initialization. (True/False) ________

5. (1 point) A ________________________ is used when you have multiple return values.

6. (4 points) Translate the following statement to an if-then-else statement:

   \[ \text{transmit} \leftarrow \text{signal}_a \text{ when state} = \text{idle} \text{ else} \]
   \[ \quad \text{signal}_b \text{ when state} = \text{incoming} \text{ else} \]
   \[ \quad \text{signal}_c \text{ when state} = \text{outgoing} \text{ else} \]
   \[ \quad \text{signal}_d; \]

7. (1 point) For every process, there is an equivalent concurrent statement. (True/False) ________

8. (4 points) (a) (2 points) Specify a CLASSIFICATION enumeration data type that spells out the
   various classifications for undergraduate students.(b) (2 points) Write a variable declaration
   MY_CLASS that has a value equal to the rightmost element of the type.

9. (1 point) Multiple architectures can exist for a single entity. (True/False) ________

10. (1 point) Multiple Choice: _____ is the default delay in VHDL. (a) Inertial (b) Transport

11. (6 points) (a) (4 points) Write a declaration of an array that can be used to hold the student numbers
    of the students in this class. (b) (2 points) Initialize the first element of this array with your student
    number.
12. (20 points) Given the following VHDL, indicate all transactions and events. Give the values of A, B, C, D, E, and F each time a change occurs. Carry this out until no further change occurs.

entity prob is
  port (D : inout bit);
end prob;

architecture PROB of PROB is
begin
  process
    A <= '1' after 5 ns,
      '0' after 12 ns;
    wait;
  end process;
  P1: process (A, C) begin
      B <= A after 2 ns;
      E <= C after 7 ns;
  end process P1;
  C <= transport A and B after 6 ns;
  P2: process (C, E) begin
      F <= C or E after 4 ns;
  end process P2;
  D <= A or B or C or F after 1 ns;
end PROB;

<table>
<thead>
<tr>
<th>Time</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ns</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5 ns</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
13. (15 points) Design a priority encoder that is described by the following truth table. (d is for don’t care)
(a) (3 points) Write a VHDL entity. (b) (6 points) Use concurrent signal assignments to implement the architecture. (c) (6 points) Use sequential statements to implement the architecture. Include any necessary library references.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>D1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
</tr>
</tbody>
</table>
ENTITY state_machine IS
  PORT (sig_in : IN BIT; clk, rst : IN BIT;
         sig_out : OUT BIT);
END state_machine;

ARCHITECTURE state_machine OF state_machine IS
  TYPE state_type IS (a, b, c, d, e);
  SIGNAL current_state, next_state : state_type;
BEGIN
  PROCESS (sig_in, current_state)
  BEGIN
    sig_out <= '0';
    next_state <= c;
    CASE current_state
    WHEN a =>
      IF sig_in = '0' THEN
        next_state <= a;
        sig_out <= '1';
      ELSE
        next_state <= d;
        sig_out <= '1';
      END IF;
    WHEN b =>
      IF sig_in = '0' THEN
        next_state <= b;
      ELSE
        next_state <= c;
      END IF;
    WHEN c =>
      IF sig_in = '1' THEN
        sig_out <= '1';
        next_state <= a;
      ELSE
        next_state <= b;
      END IF;
    WHEN d =>
      IF sig_in = '0' THEN
        next_state <= e;
      END IF;
    WHEN e =>
      IF sig_in = '1' THEN
        next_state <= c;
      END IF;
    END CASE;
  END PROCESS;
  PROCESS (clk)
  BEGIN
    IF (rst = '0') then
      current_state <= a;
    ELSIF (clk'EVENT AND clk = '1') THEN
      current_state <= next_state;
    END IF;
  END PROCESS;
END state_machine;