Part 1:

```vhdl
--Seven-segment decoder

--This is the decoder VDHL code for running a seven-segment display
--Author: Ly Nguyen
--Date: January 11, 2022

--Declaring IEEE library to be able to call a standard logic vector later
library IEEE;
use IEEE.std_logic_1164.all;

--Initializing the entity portion
ENTITY sevenseg_disp IS
   PORT( input: IN integer range 0 to 15; --declares the input for the seven segment display as an integer
         output: OUT std_logic_vector(6 DOWNTO 0) ); --declares the output for the seven segment display as 7 bits: one bit for each segment on the display
end; --declares that the entity portion is done

--Initializing the architecture portion
ARCHITECTURE encoding OF sevenseg_disp IS
   SIGNAL internal: std_logic_vector(7 DOWNTO 0); --declares SIGNAL architecture called "internal", 8 bits allow for the hex to read in two sections of 4 bits
   begin
      WITH input SELECT --declares what hex value is assigned to what value to display the hex value assignment to the seven segment display
         internal <= x"40" WHEN 00,
                     x"79" WHEN 01,
                     x"24" WHEN 02,
                     x"30" WHEN 03,
                     x"19" WHEN 04,
                     x"12" WHEN 05,
                     x"02" WHEN 06,
                     x"78" WHEN 07,
                     x"00" WHEN 08,
                     x"18" WHEN 09,
                     x"09" WHEN OTHERS; --segments 0 and 3 are off to make an H or X (however you look at it) pattern.
      output <= internal(6 DOWNTO 0);
end; --declares the end of the architecture portion
```

Part 2:

```vhdl
--Seven-segment decoder

--This is the decoder VDHL code for running a seven-segment display
--Author: Ly Nguyen
--Date: January 16, 2022

--Declaring IEEE library to be able to call a standard logic vector later
library IEEE;
use IEEE.std_logic_1164.all;

--Initializing the entity portion
ENTITY sevenseg_disp IS
   PORT( input: IN std_logic_vector(3 DOWNTO 0); --declares the input for the seven segment display as 4 bits: 0 to 3 (maximum value of 9)
         output: OUT bit_vector (6 DOWNTO 0) ); --declares the output for the seven segment display as 7 bits: one bit for each segment on the display
end; --declares that the entity portion is done

--Initializing the architecture portion
ARCHITECTURE encoding OF sevenseg_disp IS
   SIGNAL internal: bit_vector(7 DOWNTO 0); --declares SIGNAL architecture called "internal", 8 bits allow for the hex to read in two sections of 4 bits
   begin
      WITH input SELECT --declares what hex value is assigned to what value to display the hex value assignment to the seven segment display
         internal <= b"01000000" WHEN x"0",
                     b"01111001" WHEN x"1",
                     b"00100100" WHEN x"2",
                     b"00110000" WHEN x"3",
                     b"00011001" WHEN x"4",
                     b"00010010" WHEN x"5",
                     b"00000010" WHEN x"6",
                     b"01111000" WHEN x"7",
                     b"00000000" WHEN x"8",
                     b"00011000" WHEN x"9",
                     b"00001001" WHEN OTHERS; --segments 0 and 3 are off to make an H or X (however you look at it) pattern.
      output <= internal(6 DOWNTO 0);
end; --declares the end of the architecture portion
```