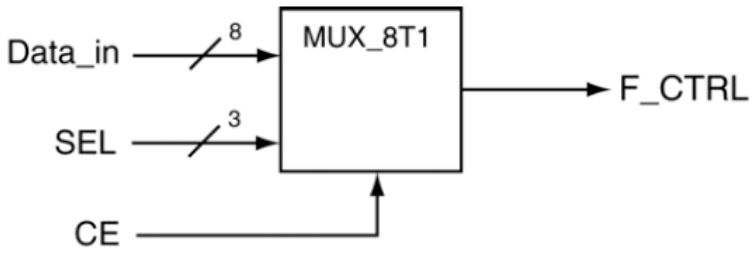
Q: Complete VHDL code that implements the 8:1 MUX shown in below.



The CE input is a chip enable. When CE = '1', the output acts like the MUX. For example, if SEL = 111 then Data_in[7] is assigned to F_CTRL and if SEL = 001 then Data_in[1] is assigned to F_CTRL. When CE is '0', the output of the MUX is '0'.

a) Use as many "**if"** statements as you deem necessary to implement your design.

```
library ieee;
use ieee.std_logic_1164.all;
entity mux_8t1_ce is
port ( Data_in : in std_logic_vector (7 downto 0);
SEL : in std_logic_vector (2 downto 0);
CE : in std_logic;
F_CTRL : out std_logic);
end mux_8t1_ce;

architecture my_8t1_mux of mux_8t1_ce is
begin
my_proc: process ( Data_in, SEL, CE  )
begin
        if (CE = '0') then
                F_CTRL <= '0';
        else
                if (SEL = "111") then F_CTRL <= Data_in(7);
                elsif (SEL = "110") then F_CTRL <= Data_in(6);
                elsif (SEL = "101") then F_CTRL <= Data_in(5);
                elsif (SEL = "100") then F_CTRL <= Data_in(4);
                elsif (SEL = "011") then F_CTRL <= Data_in(3);
                elsif (SEL = "010") then F_CTRL <= Data_in(2);
                elsif (SEL = "001") then F_CTRL <= Data_in(1);
                elsif (SEL = "000") then F_CTRL <= Data_in(0);
                else F_CTRL <= '0';
                end if;
        end if;
end process my_proc;
end my_8t1_mux;
```

b) Use "**case"** statement to implement the 8:1 MUX.

```
library ieee;
use ieee.std_logic_1164.all;
entity mux_8t1_ce is
```

```
port ( Data_in : in std_logic_vector (7 downto 0);
SEL : in std_logic_vector (2 downto 0);
CE : in std_logic;
F_CTRL : out std_logic);
end mux_8t1_ce;

architecture my_case_ex of mux_8t1_ce is
begin
my_proc: process ( SEL,Data_in, CE)
begin
        if (CE = '1') then
                case (SEL) is
                        when "000" => F_CTRL <= Data_in(0);
                        when "001" => F_CTRL <= Data_in(1);
                        when "010" => F_CTRL <= Data_in(2);
                        when "011" => F_CTRL <= Data_in(3);
                        when "100" => F_CTRL <= Data_in(4);
                        when "101" => F_CTRL <= Data_in(5);
                        when "110" => F_CTRL <= Data_in(6);
                        when "111" => F_CTRL <= Data_in(7);
                        when others => F_CTRL <= '0';
                end case;
        else
                F_CTRL <= '0';
        end if;
end process my_proc;
end my_8t1_mux;
```