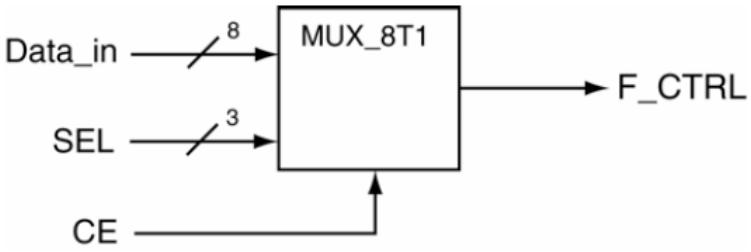
Q: Complete VHDL code that implements the 8:1 MUX shown in below.



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
```

```
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




end process my_proc;
end my_8t1_mux;
```