VHDL code for 4-bit comparator

```
--libraries to be used are specified here
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
--entity declaration with port definitions
entity compare is
port(num1 : in unsigned(3 downto 0);  --1st number
   num2 : in unsigned(3 downto 0);  --2nd number
   less : out std_logic;   -- indicates first number is smaller
   equal : out std_logic;   -- both are equal
   greater : out std_logic    -- indicates first number is bigger
);
end compare;
--architecture of entity
architecture Behavioral of compare is
begin
--Behavioral modelling to compare two 4 bit numbers.
process(num1,num2)
begin    -- process starts with a 'begin' statement
   if (num1 > num2 ) then  --checking whether num1 is greater than num2
      less <= '0';
      equal <= '0';
      greater <= '1';
   elsif (num1 < num2) then    --checking whether num1 is less than num2
      less <= '1';
      equal <= '0';
      greater <= '0';
   else     --checking whether num1 is equal to num2
      less <= '0';
      equal <= '1';
  greater <= '0';
   end if;
```

```
end process;   -- process ends with an 'end process' statement
end Behavioral;
```

Testbench code

```
--library declarations
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
--testbench entity is always empty. no ports to be declared here.
entity testbench is
end testbench;
architecture behavior of testbench is
--internal signals
signal num1,num2 : unsigned(3 downto 0) := (others => '0');
signal less,equal,greater :  std_logic:='0';
begin
--entity instantiation with named association style
uut : entity work.compare
   port map(num1 => num1,
      num2 => num2,
      less => less,
      equal => equal,
      greater => greater);
stimulus : process
begin
   --'to_unsigned' converts the integer into unsigned type
   num1 <= to_unsigned(2,4);
   num2 <= to_unsigned(9,4);
   wait for 2 ns;
   num1 <= to_unsigned(9,4);
   num2 <= to_unsigned(2,4);
   wait for 2 ns;
```

```
    num1 <= to_unsigned(10,4);
    num2 <= to_unsigned(10,4);
    wait for 2ns;
    num1 <= to_unsigned(3,4);
    num2 <= to_unsigned(6,4);
    wait for 2 ns;
    num1 <= to_unsigned(11,4);
    num2 <= to_unsigned(14,4);
    wait for 2 ns;
    num1 <= to_unsigned(16,4);
    num2 <= to_unsigned(2,4);
    wait for 2 ns;
    num1 <= to_unsigned(11,4);
    num2 <= to_unsigned(17,4);
    wait ;
    --more input combinations can be given here.
 end process stimulus;
end;
```