

<b>Course Code</b>	:	<b>BCSL-022</b>
<b>Course Title</b>	:	<b>Assembly Language Programming Lab</b>
<b>Assignment Number</b>	:	<b>BCA(2)/BCSL022/Assignment/2019-20</b>
<b>Maximum Marks</b>	:	<b>50</b>
<b>Weightage</b>	:	<b>25%</b>
<b>Last Dates for Submission</b>	:	<b>15<sup>th</sup> October, 2019 (For July 2019 Session)</b> <b>15<sup>th</sup> April, 2020 (For January 2020 Session)</b>

**This assignment has two questions of total of 40 marks. Rest 10 marks are for viva voce. Please go through the guidelines regarding assignments given in the programme guide for the format of presentation.**

**Q1.** Design a two bit counter circuit that counts from 0 to 2. It should have states 00, 01 and 10. The initial state of the counter may be assumed to be 00. The counter will be in following successive states: 00, 01, 10, 00, 01, 10, 00, 01, 10, 00 ... Use J-K flip flop to design the circuit. You must design the circuit using state transition diagram and Karnaugh's maps.

**(10)**

**Q2.** Write and run following programs using 8086 assembly language: **(3 × 10 = 30)**

- (a) Write and run an Assembly language program that converts a packed 4 digit BCD number that has been stored in two consecutive byte locations in the memory, into an equivalent binary number. The output should be stored in DX register. For example, if two consecutive byte locations have BCD values (12)<sub>h</sub> and (34)<sub>h</sub> then output will be binary equivalent of (1234)<sub>10</sub> which is (0000 0100 1101 0010)<sub>2</sub>. This binary value will be stored in DX register.
- (b) Write and run (using appropriate calling program) a near procedure in 8086 assembly language that checks if the input parameter has a value less than 5. If the value is less than 5 then subroutines displays the line "Parameter value is less than 5" else it displays "Parameter value is >= 5".
- (c) Write and run an 8086 assembly language program that finds the factorial of the value stored in BH register. You may assume that BH register will store a maximum value 8. Thus, you need not handle the overflow.