

Lab 5 Optional Questions

Flow of Control (II): Looping Statements

Question-1.

Write a program **Table.cpp** which reads a positive integer **n** and limit **m** to display a table.

Expected Output:

Example 1	Example 2
Enter a Number for Table: <u>-3</u> Enter the Limit to end the Table: <u>10</u> The Number for Table should be Positive	Enter a Number for Table: <u>3</u> Enter the Limit to end the Table: <u>-10</u> The Limit to end the Table should be Positive
Example 3	Example 4
Enter a Number for Table: <u>3</u> Enter the Limit to end the Table: <u>10</u> The Table for number 3 upto 10 is 1 * 3 = 3 2 * 3 = 6 3 * 3 = 9 4 * 3 = 12 5 * 3 = 15 6 * 3 = 18 7 * 3 = 21 8 * 3 = 24 9 * 3 = 27 10 * 3 = 30	Enter a Number for Table: <u>5</u> Enter the Limit to end the Table: <u>10</u> The Table for number 5 upto 10 is 1 * 5 = 5 2 * 5 = 10 3 * 5 = 15 4 * 5 = 20 5 * 5 = 25 6 * 5 = 30 7 * 5 = 35 8 * 5 = 40 9 * 5 = 45 10 * 5 = 50

Question-2.

Write a program **Prime.cpp** which reads a positive integer **n** and prints whether it is prime or composite. A number **n** is considered as prime if it has only two factors, for example, **n=5** is prime because it is divisible by only two factors 1 and 5 (itself).

Expected Output:

Example 1	Example 2
Enter a Positive Number: <u>0</u> 0 is not a positive number.	Enter a Positive Number: <u>-5</u> -5 is not a positive number.

Example 3	Example 4
Enter a Positive Number: <u>5</u> 5 is a prime number because it has only two factors 1 and 5.	Enter a Positive Number: <u>6</u> 6 is a composite number.
Example 5	
Enter a Positive Number: <u>1</u> 1 is neither prime nor composite.	