LAB SHEETS
ON
C PROGRAMMING

LAB 1

FORMATTED AND UNFORMATTED I/O IN C

Task #1: WAP in C to input a character from the user and display the entered character using the concept of getchar() and putchar()

Task #2: WAP in C to input a string from the user and display the entered string using the concept of gets() and puts()

Task #3: WAP in C to input name, roll, marks and address of a student from the user and display the entered details using the concept of scanf() and printf()

Task #4: WAP in C to convert a temperature from degree Celsius to degree Fahrenheit.

Task #5: WAP in C to add two complex numbers

LAB 2

CONTROL STATEMENTS IN C PART I

Task #1: WAP in C to find the greatest among three numbers entered by the user

Task #2: WAP in C to find the roots of a quadratic equation $ax^2 + bx + c = 0$, whose coefficients a, b and c are entered by the user.

Task #3: WAP in C to display multiplication table of a number entered by the user

Task #4: WAP in C to determine whether a number entered by the user is prime or composite.

Task #5: WAP in C to convert a decimal number into its binary equivalent.

LAB 3

CONTROL STATEMENTS IN C PART II

Task #1: WAP in C to generate the following pattern:

```
1
12
123
1234
12345
```

Task #2: WAP in C to generate the following pattern:

```
1
22
333
4444
55555
```

Task #3: WAP in C to generate the following pattern:

```
1 2 3 4 5 4 3 2 1
  1 2 3 4 3 2 1
    1 2 3 2 1
      1 2 1
        1
```

Task #4: WAP in C to display the following pattern:

```
          *
        * * *
      * * * * *
    * * * * * * *
* * * * * * * * *
    * * * * * * *
      * * * * *
        * * *
          *
```

Task #5: WAP in C to display the following pattern:

```
1 2   3
2 4   6
3 6   9
```

LAB 4

ARRAYS IN C PART I

Task #1: WAP in C to sort n numbers entered by the user in ascending order

Task #2: WAP in C to display largest, smallest, and third largest among n numbers entered by the user

Task #3: WAP in C to display transpose of a mxn matrix entered by the user

Task #4: WAP in C to count the total no. of words and characters in a line of string entered by the user.

LAB 5

ARRAYS IN C PART II

Task #1: WAP in C to find the product of two 3 x 3 matrices entered by the user

Task #2: WAP in C to find the sum of diagonal elements in a 3 x 3 matrix entered by the user

Task #3: WAP in C to check whether a string entered by the user is palindrome or not.

Task #4: WAP in C to sort the names of *n* employees in ascending order (alphabetic order).

.

LAB 6

FUNCTIONS IN C

Task #1: WAP in C to add two numbers using

a) Function with return type and argument
b) Function with return type and no argument
c) Function with no return type and argument
d) Function with no return type and no argument

Task #2: WAP in C to convert a decimal no. into its binary equivalent using function

Task #3: WAP in C to check whether a no. entered by the user is prime or composite using function

Task #4: WAP in C to calculate the factorial of a no. entered by the user using recursive function

Task #5: WAP in C to find the sum of squares of first *n* natural numbers using recursive function.

LAB 7

STRUCTURE AND UNION IN C

Task #1: WAP in C to input name, roll, marks and address of a student and display the entered details using the concept of structure

Task #2: WAP in C to input name, roll, marks and address of *n* students and display the entered details using the concept of an array of structure.

Task #3: WAP in C to input name, roll, marks and address of *n* students and display the entered details in the order of marks.

Task #4: WAP in C to input name, roll, marks and address of a student and display the entered details using the concept of union

LAB 8

POINTERS IN C

Task #1: WAP in C input *n* numbers from the user and display the entered numbers using the concept of pointer.

Task #2: WAP in C to sort *n* numbers entered by the user in ascending order using the concept of pointer.

Task #3: WAP in C find the sum and average of *n* numbers entered by the user using the concept of Dynamic Memory Allocation.

Task #4: WAP in C to sort *n* numbers entered by the user in ascending order using the concept of Dynamic Memory Allocation

LAB 9

FILE HANDLING IN C

Task #1: WAP in C to write characters to a text file student.txt until the user press enter key.

Task #2: WAP in C to read the file student.txt created in the previous task.

Task 3: WAP in C to copy a file source.txt into destination.txt

Task #4: WAP in C to write the name, roll, marks and address of *n* students to a data file using the concept of ***fprintf***().

Task #5: WAP in C to read the file created in the previous task using the concept of ***fscanf***()

Task #6: WAP in C to read a file num.txt containing integers. Write all the prime numbers from the file num.txt to another file prime.txt.

LAB 10

GRAPHICS IN C

Task #1: WAP in C to draw a line from (200, 250) to (300, 350)

Task #2: WAP in C to draw a circle with center at (250,250) and radius 100.

Task #3: WAP in C to draw an ellipse of your choice with yellow color.

Task #4: WAP in C to draw a rectangle of your choice with red color.

Task #5: WAP in C to draw an arc of your choice with blue color.

***Task #6*:** WAP in C to output a string entered by the user starting at co-ordinate position (200,250).