Algorithms.

Steps involving in algorithm, assignment, function call, arithmetic operations, variable declarations, conditional and looping statements.

Worst case, average case, best case.

Time and space complexity, Big Oh, Big Theeta, Big Omega notations.

String / character arrays

1. WAP to check whether an input character is alphabet or not?
2. WAP to check whether a triangle is valid or not by providing all the sides.
3. WAP to input three side of a triangle and check whether the triangle is equilateral, isosceles, or scalene

Series printing and sum of series problem.

$S=1+2+3+4+\dots\dots$
$S=2+4+6+8+\dots.$
$S=1+3+5+7+\dots.$
$S=2+6+12+20+\dots.$
$S=0+6+24+60+\dots.$
$S=1+2!+3!+4!+\dots.$
$S=1+2^2+3^3+4^4+\dots$
$S=1+2^{/}2!+3^{/}3!+4^{/}4!+\dots$
$S=1+2^{2/}2!+3^{3/}3!+4^{4/}4!+\dots$
$S=1+2^{/}2!+3^{2/}3!+4^{3/}4!+\dots$
$S=1-2!+3!-4!+5!-6!+\dots$

Pattern printings problem

```
*          1          1          A               E
* *        1 2        2 2        B B             D D
* * *      1 2 3      3 3 3      C C C           C C C
* * * *    1 2 3 4    4 4 4 4    D D D D         B B B B
* * * * *  1 2 3 4 5  5 5 5 5 5  E  E  E  E E    A A A A A

* * * * *  5 5 5 5 5  1 2 3 4 5  5 4 3 2 1  A B C D E  A A A A A
* * * *    4 4 4 4    1 2 3 4    4 3 2 1    A B C D    B B B B
* * *      3 3 3      1 2 3      3 2 1      A B C      C C C
* *        2 2        1 2        2 1        A B        D D
*          1          1          1          A          E


1               A                     E
1 2 1           A B A                 E D E
1 2 3 2 1       A B C B A             E D C D E
```

```
1 2 3 4 3 2 1          A B C D C B A              E D C B C D E
1 2 3 4 5 4 3 2 1      A B C D E D C B A          E D C B A B C D E
```

```
        *                 A                5
      *   *             B   B            4   4
    *   *   *         C   C   C        3   3   3
  *   *   *   *     D   D   D   D    2   2   2   2
*   *   *   *   *  E   E   E   E   E 1   1   1   1   1
```

```
*   *   *   *   *  E   E   E   E   E  1   1   1   1   1
  *   *   *   *     D   D   D   D      2   2   2   2
    *   *   *         C   C   C          3   3   3
      *   *             B   B              4   4
        *                 A                  5
```

```
        *                    1                          E
       * *                 1 2 1                      E D E
      * * *              1 2 3 2 1                  E D C D E
     * * * *           1 2 3 4 3 2 1              E D C B C D E
    * * * * *        1 2 3 4 5 4 3 2 1          E D C B A B C D E
     * * * *           1 2 3 4 3 2 1              E D C B C D E
      * * *              1 2 3 2 1                  E D C D E
       * *                 1 2 1                      E D E
        *                    1                          E
```

```
        *                    5                          A
       * *                 5 4 5                      A B A
      * * *              5 4 3 4 5                  A B C B A
     * * * *           5 4 3 2 3 4 5              A B C D C B A
    * * * * *        5 4 3 2 1 2 3 4 5          A B C D E D C B A
     * * * *           5 4 3 2 3 4 5              A B C D C B A
      * * *              5 4 3 4 5                  A B C B A
       * *                 5 4 5                      A B A
        *                    5                          A
```

#Prime number (first n prime number and prime numbers upto n)

#Fibonacci series (first n terms / terms upto n), to check a number is Fibonacci term or not.

HCF & LCM

Reverse of a number/ digits count / even & odd digits and their sum respectively / Armstrong number / palindrome number / strong numbers / converts a numbers into characters, print digits of a number in words, Magic number.

Write C programs for number system conversion

1). Decimal to others

2). Others to decimal

3). Others to others

Bitwise operators (& , | , ~ , ^ , << , >>), their examples, hexadecimal and octal numbers representation in C.

ARRAYS

- Sum of n numbers, largest numbers, smallest numbers, average of n numbers, their address, unique numbers in an array, duplicate numbers in an array, frequency of each element in an array,
- Write a program in C to separate odd and even integers in separate arrays
- Write a program in C to count a total number of duplicate elements in an array.
- Write a program in C to print all unique elements in an array.
- Write a program in C to count the frequency of each element of an array
- Write a program in C to find the second largest element in an array.
- Write a program in C to find a pair with given sum in the array
- Write a program in C to find the number of times (frequency) occurs a given number in an array
- Write a program in C to rearrange positive and negative numbers alternatively in a given array
- Write a program in C to check whether an array is subset of another array.
- Write a program in C to rearrange an array in such an order that– smallest, largest, 2nd smallest, 2nd largest and on

- 2D Array, matrix using 2D array, various problem in matrix (lower triangle/ upper triangle/ transpose of matrix/ matrix multiplications) .
- Write a program in C to calculate determinant of a 3 x 3 matrix
- Write a program in C to accept a matrix and determine whether it is a sparse matrix
- Write a program in C to check whether a given matrix is an identity matrix
- Searching and sorting in arrays using bubble, selection, and insertion sort. Merging of two sorted array.

STRINGS

Initializing a character array (string).

Input strings using scanf(), gets(), fgets() and character by character using loop (scanf() and getche()).

Output string using printf(), puts, character by character (using loop).

String handling functions (strcpy(), strcat(), strlen(), strrev(), strcmp()).

- WAP to implements all the above string handling functions manually.
- WAP to check whether a string is palindrome or not using string function and without using string function.
- WAP to count alphabet, vowels, consonants, digits, special characters in a given string.
- WAP to convert a string into uppercase and vice versa.

- WAP to count occurrence/frequency of all the characters in a string.
- WAP to create of powerset of all the .
- WAP to take N names from user as input and sort then in ascending order.
- WAP to search occurrence of a substring within a string.
- WAP to search occurrence of a substring within a string and replace it with another string.

FUNCTIONS

- Write a program in C to show the simple structure of a function.
- WAP in C to swap two numbers using function.
- WAP in C to show the passing arguments in a function as call by value and call by address.
- WAP in C to find the GCD of N numbers using function.
- WAP in C to find the factorial of a given number using recursive function.
- WAP in C to find the GCD of two numbers using recursion.
- WAP in C to print the first N terms of Fibonacci series using recursion.
- WAP to show how to pass the array in a function and return as well.
- WAP to print an array using recursion.
- WAP to pass and handle the command line arguments to the program.

STRUCTURE

- WAP to create a structure Student with (name, subject, roll, sid, marks) using appropriate data types, then take input and display the values of member variable of structure variable.
- Create array of structure of above data type and perform the I/O operations for N number of students.
- WAP to pass structure variable as parameters into function as well as return value of a structure variable from the called function.
- WAP to demonstrate how to pass array of structure as parameters into a function as well as return array of structure from the called function.
- WAP to print an array of Student structure (mentioned above) by using structure pointer.
- WAP to allocate memory of a user defined structure size dynamically. Perform I/O operations.

Self-referential structure

Implementing single linked list using dynamic memory allocation.

- WAP to implement Single linked list using the following menu driven functions.
  1. Insert at Beginning
  2. Insert at End
  3. Insert at specific position
  4. Display
  5. Delete
  6. Reverse Display
  7. Reverse the linked list
  8. Search
  9. Sort (using selection sort)

FILE HANDLING

- WAP to show how to open and close a file for reading as well as writing purpose.
- WAP to read entire data from an existing file in character by character manner using **fgetc()** method.
- WAP to read entire data from an existing file using string by **fgets()** method.
- WAP to write data to a file in character-by-character manner.
- WAP to write data to a file in string format.
- WAP to implement formatted Input/Output using various file handling methods.
- WAP to demonstrate how to write a structure variable into a file, also show how to read the same from file as well.
- WAP to display all the contents of an existing text file in reverse order.
- WAP to copy contents of a file into another file.
- WAP to create command '**merge**' to concatenate contents of a file into another file.

  Syntax : merge <first file name> <second file name>
- WAP to search and replace a string in an existing file with another string.
- Write a menu driven program to implement a file-based student's data base with the following options.
  1. Insert
  2. Display
  3. Delete
  4. Search
  5. Edit
  6. Sort