C Language Practical’s

1. Introduction of C

a. Print Hello, World!
b. Write a Program for Simple Input/Output
c. Input Two Numbers From User and make Sum of Two Numbers
d. Write a Program to Character Input/Output
e. Write a Program to Floating Point Input/Output

2. **Control Structures** : Control structures in C are used to control the flow of execution of the program. This includes conditional statements (if, else, switch) and loops (for, while, do-while).

1. If-Else Condition :

a. Check if a number is positive or negative
b. Check if a number is even or odd
c. Find the largest of two numbers
d. Check if a character is a vowel or consonant
e. Check if a character is an alphabet
f. Check if a number is divisible by both 5 and 11
g. Check if a student passed based on marks
h. Check if a triangle is valid based on angles

2. Nested If-Else

a. Check if a number is positive, negative, or zero
b. Check if a year is a leap year
c. Find the largest of three numbers
d. Check if a Number is Divisible by 3 and 5
e. Determine the Grade Based on Marks
f. Determine the Largest of Three Numbers
g. Check if a Number is Positive, Negative, Zero, or Even/Odd

3. Switch case

a. Simple Calculator (Addition, Subtraction, Multiplication, Division)
b. Determine the Day of the Week from a Number (1-7):
c. Menu Selection of Hotel Food
d. Temperature Converter
e. Basic ATM Menu

C Language Practical’s

4. For Loop
    a. Print Numbers from 1 to 10
    b. Print the Sum of Numbers from 1 to 10
    c. Print the Factorial of a Number
    d. Print Multiplication Table of a Number
    e. Print the Fibonacci Sequence up to N Terms
    f. Print Prime Numbers between 1 and 100
    g. Printf All Patterns Given last of the report
    h. Print the First N Natural Numbers in Reverse Order
    i. Calculate the Sum of Even Numbers between 1 and N
    j. Print the Sum of Digits of a Number

5. While Loop
    a. Write all programs again using while loop

6. do…While Loop
    a. just change some portion of all while loop programs and implement do…while loop

Array : Arrays in C are collections of elements of the same type, stored in contiguous memory locations.

7. One Dimensional Array Programs
    a. Take input numbers and print it.
    b. Take input String and print it.
    c. Take Input Elements and Sum of Array Elements
    d. Find the Maximum Element
    e. Find the Minimum Element
    f. Reverse Array
    g. Copy Array
    h. Find the Average of Array Elements
    i. Find the Index of a Specific Value
    j. Sort Array in Ascending Order

8. Two Dimensional Array Programs
   a. Print a 2D Array
   b. Sum of All Elements in a 2D Array
   c. Find the Maximum Element in a 2D Array
   d. Transpose of a 2D Array
   e. Multiply Two 2D Arrays
   f. Add Two 2D Arrays
   g. Row-wise Sum of a 2D Array
   h. Check if a Matrix is Symmetric

Function : Functions in C are blocks of code that perform a specific task. They allow for code reuse and better organization.

9. Programs using Function
   a. Print Hello World Using Function
   b. Print sum of two numbers using function of return value
   c. Write example of Void function
   d. Function with Multiple Return Values Using Pointers
   e. Write a recursive program of factorial using function
   f. Print Fibonacci Series using recursive function
   g. Write an example of Function overloading
   h. Return an array using function
   i. Function with Struct Parameter

C Language Practical's

Pointers : Pointers in C are variables that store the memory address of another variable. They are used for dynamic memory allocation, array manipulation, and function argument passing by reference.

10. Pointer Declaration and Derefrencing
11. Pointer Arithmetic
12. Pointer to pointer
13. Pointer as function Argument

Structures : Structures in C are user-defined data types that group related variables of different data types. They are used to represent a record.

14. Structure Declaration and Initialization
15. Array of Structures
16. Nested Structures
17. Structure with Function

File management : File management in C involves creating, reading, writing, and closing files using functions provided by the stdio.h library.

18. File Creation and writing
19. File Reading
20. File Append
21. File Copy
22. File Delete

Dynamic Memory Allocation : Dynamic memory allocation in C involves allocating and freeing memory at runtime using functions such as malloc, calloc, realloc, and free.

23. Malloc Example
24. Calloc Example :
25. Realloc Example:
26. Dynamic Array :
27. Dynamic 2D Array

String Handling : String handling in C involves manipulating and working with arrays of characters. The string.h library provides various functions for string operations.

28. String Input/Output
29. String Length
30. String Copy
31. String Concatenation

C Language Practical’s

32. String Comparison