Unit - I

Introduction to Problem Solving and Programming Concepts

SR. No.	Question	BTL
1.	Explain the difference between primary and secondary memory	BTL2
2.	Construct an algorithm to calculate the factorial of a number.	BTL3
3.	Develop flowchart to determine if the user entered number is an Armstrong number or not.	BTL3
4.	Explain the key characteristics of the each generation of computers	BTL2
5.	What is the role of the CPU in a computer's architecture?	BTL1
6.	List and explain steps present in the problem-solving process?	BTL2
7.	Construct an algorithm and pseudocode to swap two values without using a third variable.	BTL3

8.	Develop flowchart to determine if the user entered number is an Armstrong number or not.	BTL3
9.	Develop flowchart to find number is even or odd.	BTL3
10.	Develop algorithm and pseudocode to calculate area and perimeter of rectangle.	BTL3
11.	Compare between System Software and application software	BTL2
12.	Develop flowchart to calculate average of the first N natural numbers.	BTL3
13.	Compare Machine Level Language and High Level Language.	BTL2
14.	What are the steps involved in executing a program?	BTL1
15.	What is a logic error, and why is it often difficult to detect?	BTL1
16.	What are the main components of a C program?	BTL1
17.	Explain the significance of assembler, interpreter and compiler.	BTL2

18.	Develop algorithm and flowchart to calculate simple interest and compound interest.	BTL3

UNIT-II : Programming Basics and Control Structures

Q. No.	Questions	BTL
1.	What are Looping statements? Explain any two with syntax and example.	BTL2
2.	Construct a C program to find the biggest of three numbers.	BTL3
3.	Explain switch statement with syntax and example.	BTL2
4.	Explain with syntax the different loops used in C program.	BTL2
5.	Develop a program in C to find the sum of n natural numbers.	BTL3
6.	Develop a C program to find GCD of two numbers.	BTL3
7.	Develop a C program to determine if the user entered number is a prime number or not.	BTL3
8.	What are looping statements in C programming? Explain any two with syntax and examples.	BTL2
9.	Distinguish between break and continue statements with an example	BTL4
10.	Develop a calculator program in C language to do simple operations like addition, subtraction, multiplication and division. Use switch statement in your program.	BTL4

11.	Develop a program to check whether the person is eligible to vote.	BTL4
12.	Define goto with an example.	BTL1
13.	Define exit and return statements.	BTL1
14.	Construct a program to generate prime numbers between 1 and n	BTL3
15.	Construct a program to find the sum of the individual digits of a given number.	BTL3
16.	Construct a program to find the sum of even and odd numbers from 1 to n.	BTL3
17.	What is a nested loop? Construct a program to display multiplications tables from 1 to n.	BTL3
18.	Construct a program to display the following pattern. ***** **** *** ** *	BTL3

Unit - III

Array and Strings

SR. No.	Question	BTL
1.	Explain the syntax for declaring a one-dimensional array in C.	BTL2
2.	How do you declare a two-dimensional array? Provide an example.	BTL1
3.	How can you loop through all elements of a one-dimensional array? Provide example.	BTL1
4.	Explain how to access elements in a two-dimensional array using indices.	BTL2
5.	Construct a program that finds the maximum element in a one-dimensional array.	BTL3
6.	How can you reverse the elements of a one-dimensional array in place?	BTL1
7.	What is a two-dimensional array? Give an example.	BTL1

8.	How does the strlen() function work? Provide an example.	BTL1
9.	Develop a program to count the number of vowels in a given string using arrays.	BTL3
10.	Develop a program that reverse the elements of a one-dimensional array.	BTL3
11.	What is an array? How is it different from a variable?Explain with an example.	BTL1
12.	Construct a C program that finds the largest element in a one-dimensional array.	BTL3
13.	Develop a program that uses strcat() to concatenate two strings.	BTL3
14.	Explain with a suitable example, how to access elements in a two-dimensional array using indices.	BTL1
15.	Distinguish between one-dimensional array and two-dimensional array with a suitable example.	BTL3
16.	Construct a program that finds the smallest element in a one-dimensional array.	BTL3

Unit - IV

Functions and Pointers

SR. No.	Question	BTL
1.	Explain the scope of a variable in a function? Explain local and global scope.	BTL2
2.	What is meant by the order of passing arguments in a function call?	BTL1
3.	What is the purpose of the return statement in C? Write a simple program to demonstrate its use.	BTL1
4.	Distinguish between call by reference and call by value with suitable example.	BTL3
5.	Build a function is_leap_year which takes the year as its argument, checks whether the year is a leap year or not, and then displays appropriate message on the screen.	BTL3
6.	Define pointers, explain the difference between a null pointer and a void pointer.	BTL1
7.	Develop a program to calculate the factorial of a number using the user defined function "*fact*".	BTL3

8.	Analyze how a program behaves when the number of actual parameters is less than the formal parameters in a function. Identify possible errors or outcomes, explain with an example.	BTL3
9.	Why are functions needed in C? Explain the concept of a function, its syntax, and provide a suitable example.	BTL3
10.	Distinguish between Library functions and User defined functions in C programming and explain with examples.	BTL1
11.	Build a program that uses functions to calculate the area and circumference of a circle. The radius should be passed as a parameter.	BTL3
12.	Explain recursion with a suitable example.	BTL2
13.	Explain how to use functions like malloc, calloc, and free in C for dynamic memory management.	BTL2
14.	Develop C program for recursive function to calculate the factorial of a number.	BTL3
15.	What are pointer expressions? How can they be used in arithmetic operations?	BTL1
16.	Develop C program to calculate the average of first n numbers using function.	BTL3

17.	What are advantages and disadvantages of call by reference technique?	BTL1
18.	Develop C program to check year is leap year or not using function.	BTL3

UNIT- V : Structures and Unions

Q. No.	Questions	BTL
1.	What are the storage classes available in C?	BTL1
2.	Define Macro in C.	BTL1
3.	Define Union in C , Explain how members of a union are accessed using a program code?	BTL1
4.	Develop a structure named "Car" to store details like car ID, model, and cost. Write a C program to input data for three cars, and display the information.	BTL3
5.	Distinguish between a structure and an array with suitable example.	BTL4
6.	Explain the concept of nested structures with an example.	BTL1
7.	Develop a structure named Book to store book details like title, author, and price. Write a C program to input details for three books, find the most expensive and the lowest priced books, and display their information.	BTL3
8.	Distinguish between structure and union, Explain in which applications union can be useful?	BTL4
9.	What do you mean by structure definition? Define Structure "Employee" to store Employee name, Designation, Date of Joining, Salary.	BTL3
10.	Develop a C program to define a structure to store the information of 5 students, including their name, roll number, and marks. Use an array of structures and print the details of all students	BTL4

11.	Distinguish between structure and union with suitable example.	BTL1
12.	What is a structure? Create a structure with data members of various types and declare two structure variables.	BTL1
13.	Construct a program to read data into these and print the same. Justify the need for structured data type.	BTL3
14.	Define a structure named Time with members hours, minutes, and seconds. Construct a C program to input two times, add them, and display the result in proper time format.	BTL3
15.	Define a structure named "Date" with members day, month, and year. Write a C program to input two dates and find the difference in days between them.	BTL3

Unit - VI

File Operations

SR. No.	Question	BTL
1.	What function would you use to read an entire line from a file in C?	BTL1
2.	Explain the difference between fgets( ) and gets( ) with suitable example.	BTL1
3.	Develop a program to read and display the contents of a text file.	BTL3
4.	Develop a program to count characters, tabs and spaces in a text file	BTL3
5.	Explain the purpose of each mode used for opening a file?	BTL1
6.	Compare different standard library functions for reading data from a file with suitable example.	BTL3
7.	Build a C program to append the contents of "*try1.txt*" at the end of another file "*try2.txt*".	BTL4
8.	Explain the syntax for opening a file in C with a suitable example. What parameters does it require?	BTL1
9.	Develop a program that counts the number of characters in a file.	BTL3

10.	Distinguish between reading characters and reading strings from a file.	BTL4
11.	What are some common reasons for failure when attempting to open a file?	BTL1
12.	How can you handle errors when a file fails to open? Provide a code snippet demonstrating this.	BTL1
13.	What is the importance of checking if a file opened successfully?	BTL1
14.	Why is it important to close a file after operations are complete?	BTL1
15.	What function is used to close a file in C? What does it return?	BTL1
16.	What is the effect of opening a file in "append" mode versus "write" mode?	BTL1
17.	How do you write a string to a file? Give a code example.	BTL1
18.	Explain some real-life applications where file handling is essential.	BTL2