

SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur – 603 203

DEPARTMENT OF INFORMATION TECHNOLOGY

QUESTION BANK

V SEMESTER



IT3563 – C PROGRAMMING AND DATA STRUCTURES

Regulation – 2023

Academic Year 2026 – 2027 (ODD)

Prepared by

Mr.M.ASAN NAINAR, AP



SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)
SRM Nagar, Kattankulathur – 603 203.

DEPARTMENT OF INFORMATION TECHNOLOGY



QUESTION BANK

SUBJECT : IT3563 – C PROGRAMMING AND DATA STRUCTURES

YEAR/SEM : III Year / V Semester

UNIT I C PROGRAMMING FUNDAMENTALS			
Data Types – Variables – Operations – Expressions and Statements – Conditional Statements – Functions – Recursive Functions – Arrays – Single and Multi-Dimensional Arrays.			
PART – A			
Q.No	Questions	BT Level	Competence
1	List data types available in “C”.	BTL1	Remember
2	Justify why keywords are not to be used as identifiers.	BTL2	Understand
3	Differentiate between variable and constant.	BTL2	Understand
4	State enumeration constant.	BTL1	Remember
5	Distinguish between Expression and statements.	BTL2	Understand
6	Write the use of ternary operator?	BTL1	Remember
7	What is a Modulo Operator? What is the use of sizeof() operator?	BTL1	Understand
8	Mention the difference between ++a and a++?	BTL1	Remember
9	What are the I/O Functions in C?	BTL2	Understand
10	List out the Escape Sequences present in “C”.	BTL1	Remember
11	Differentiate scanf() and gets() function?	BTL2	Understand
12	Give an example for if and if-else conditional statements.	BTL1	Remember
13	Differentiate Entry and Exit controlled constructs.	BTL2	Understand
14	Define function with its syntax.	BTL1	Remember
15	Difference between user defined and predefined functions.	BTL2	Understand
16	Why scope of variables necessary in function?	BTL2	Understand
17	Express the difference between function declaration and function definition.	BTL2	Understand
18	What is meant by Recursive function? Give an example.	BTL1	Remember
19	Write a program to calculate GCD using a recursive function?	BTL1	Remember
20	State Static and external Storage Class.	BTL2	Understand
21	What is an array? Give an example.	BTL1	Remember
22	Write the features of the array.	BTL1	Remember
23	How is an array represented in memory?	BTL2	Understand
24	How to create a two-dimensional array?	BTL2	Understand

PART-B			
1	(i) Explain in detail about “C” declarations and variables. (8) (ii) What are constants? Explain the various types of constants in C. (8)	BTL3	Apply
2	Discuss about the various data types in “C”.	BTL4	Analyze
3	Explain about the various decision making statements in “C” language.	BTL3	Apply
4	Explain various operators in c with examples.	BTL3	Apply
5	Explain briefly about the formatted and unformatted I/O function in C.	BTL3	Apply
6	What is the purpose of a looping statement? Explain in detail the operations of various looping statements with examples.	BTL4	Analyze
7	(i) Write a C program to check whether the given number is palindrome or not. (8) (ii) Write a C program to sum of digits of an integer. (8)	BTL4	Analyze
8	(i) Write a program to find whether a number is prime or not. (8) (ii) Write a C program to find the largest among 3 numbers entered by the user.(8)	BTL4	Analyze
9	Describe the structure of a C program using “Calculator program” example.	BTL4	Analyze
10	(i) Explain in detail about function prototype. (8) (ii) What is a storage class? Explain its types with examples.(8)	BTL4	Analyze
11	Explain Recursive function with suitable examples.	BTL3	Apply
12	Explain the concept of call by value and call by reference in detail.	BTL3	Apply
13	What is an array? Explain the characteristics and Classification of an array with examples.	BTL4	Analyze
14	(i) Write a C program to find the largest and smallest element in an array. (8) (ii) Write a C program to search a given number in an array of elements. (8)	BTL4	Analyze
15	Write a program for multiplication of two matrices.	BTL4	Analyze
16	Write a program for one dimensional array.	BTL4	Analyze
17	Write a C program for addition of two matrices and display the result.	BTL3	Apply

UNIT II C PROGRAMMING – ADVANCED FEATURES			
Structures – Union – Enumerated Data Types – Pointers: Pointers to Variables, Arrays and Functions – File Handling – Preprocessor Directives.			
PART – A			
Q.No	Questions	BT Level	Competence
1	What is a structure in C? Give an example.	BTL1	Remember
2	How do you declare and access members of a structure?	BTL1	Remember
3	State the use of typedef with structures.	BTL2	Understand

4	Differentiate between an array and a structure.	BTL2	Understand
5	Define union in C? How is it different from a structure?	BTL2	Understand
6	How much memory is allocated to a union? Explain.	BTL2	Understand
7	Difference between structure and union.	BTL2	Understand
8	What is an enumerated data type in C?	BTL1	Remember
9	What is meant by pointer? How do you declare it?	BTL1	Remember
10	What are * and & operators means?	BTL1	Remember
11	How do you use pointers to access array elements?	BTL2	Understand
12	State dangling pointer.	BTL1	Remember
13	What do you mean by array of pointers?	BTL2	Understand
14	How do you use a pointer to a function?	BTL2	Understand
15	What is a NULL pointer? Why is it important?	BTL2	Understand
16	List two advantages of using function pointers in C programming.	BTL2	Understand
17	List out the basic file operations in C.	BTL1	Remember
18	Difference between text and binary files in C.	BTL2	Understand
19	How do you open a file in read and write mode?	BTL2	Understand
20	What are preprocessor directives? Give two examples.	BTL1	Remember
21	Differentiate between #define and #include.	BTL2	Understand
22	Illustrate the purpose of fclose() in file handling.	BTL2	Understand
23	What is the use of pre-processor directives?	BTL2	Understand
24	List some preprocessor directives in C.	BTL1	Remember

PART-B

1	What is a structure in C? Write a program to define and display student details using a structure.	BTL3	Apply
2	Can a structure contain another structure? Explain with a simple program.	BTL3	Apply
3	Write a C program to define a structure for a student with fields: name, ID, and marks. Input and display data.	BTL3	Apply
4	Create a union to store an integer, a float, and a character. Show how memory is shared.	BTL3	Apply
5	Design a C program that manages employee records using structures and pointers.	BTL6	Create
6	What is a union in C? How does it differ from a structure?	BTL4	Analyze
7	What is an enumerated data type (enum) in C? Show how to declare and use it.	BTL3	Apply
8	What is a pointer in C? How do you declare and use it to access a variable?	BTL3	Apply
9	Write a program to swap two numbers using pointers.	BTL3	Apply
10	How do you use a pointer to an array? Show with a simple example.	BTL3	Apply
11	What are function pointers? Write a program to call a function using a function pointer(Call by Reference).	BTL4	Analyze
12	Explain file handling in C? Why is it useful?	BTL4	Analyze

13	Write a program to write and read a string from a file.	BTL3	Evaluate
14	Explain the difference between text and binary file modes.	BTL4	Analyze
15	Explain in detail about preprocessor directives with examples.	BTL3	Apply
16	Explain the use of #define in C with a simple program.	BTL3	Apply
17	Write a program using enum to display the value of days like MONDAY = 0, TUESDAY = 1, etc.	BTL3	Apply

UNIT III LINEAR DATA STRUCTURES

Abstract Data Types (ADTs) – List ADT – Array-Based Implementation – Linked List – Doubly- Linked Lists – Circular Linked List – Stack ADT – Implementation of Stack – Applications – Queue ADT – Priority Queues – Queue Implementation – Applications.

PART – A

Q.No	Questions	BT Level	Competence
1	Define Abstract Data Type (ADT).	BTL1	Remember
2	What is a linked list?	BTL1	Remember
3	Define stack with suitable examples.	BTL1	Remember
4	What is a queue? Give an example.	BTL1	Remember
5	Define circular linked list.	BTL1	Remember
6	What is the difference between static and dynamic memory allocation?	BTL2	Understand
7	List two applications of stack.	BTL1	Remember
8	Define priority queue.	BTL1	Remember
9	State the use of rear and front in a queue.	BTL2	Understand
10	What is a node in a linked list?	BTL1	Remember
11	Difference between singly and doubly linked lists.	BTL2	Understand
12	Write the syntax for declaring a structure in C.	BTL1	Remember
13	Mention the concept of list ADT.	BTL2	Understand
14	Determine array-based implementation of a list.	BTL2	Understand
15	How insertion is performed in a singly linked list?	BTL2	Understand
16	Compare stack and queue.	BTL2	Understand
17	Differentiate between linear and circular queue.	BTL2	Understand
18	List the term LIFO with respect to stacks.	BTL1	Remember
19	List out the enqueue operation in a queue.	BTL1	Remember
20	How does a circular linked list differ from a regular linked list?	BTL2	Understand
21	Why do we use doubly linked lists instead of singly linked lists?	BTL2	Understand
22	How can a stack be implemented using arrays?	BTL2	Understand
23	List out the real-time application of queue.	BTL1	Remember
24	How is memory managed in linked list implementation?	BTL2	Understand

PART-B			
1	Write a program to implement a singly linked list with insertion and deletion operations.	BTL3	Apply
2	Implement stack operations using arrays in C.	BTL3	Apply
3	Write a program to insert and delete elements in a queue using linked list.	BTL3	Apply
4	Implement a circular linked list and perform insert at beginning and end.	BTL3	Apply
5	Develop a C program to reverse a string using stack.	BTL3	Apply
6	Analyze the time and space complexity of array-based vs linked list-based list implementations.	BTL4	Analyze
7	Compare stack implementation using array and linked list.	BTL4	Analyze
8	Analyze the differences between queue and priority queue with examples.	BTL4	Analyze
9	Explain the differences between linear, circular, and doubly linked lists using diagrams and code snippets.	BTL4	Analyze
10	Evaluate the performance of stack in recursive function calls.	BTL5	Evaluate
11	Justify the use of circular queue over linear queue in memory-constrained environments.	BTL5	Evaluate
12	Evaluate the role of pointers in efficient memory management in linked list implementations.	BTL5	Evaluate
13	Discuss the pros and cons of using array-based stacks in embedded systems.	BTL5	Evaluate
14	Design a menu-driven program to manage a student record system using structures and linked lists.	BTL6	Create
15	Create a priority queue system for a hospital using structures and linked lists.	BTL6	Create
16	Develop a stack-based expression evaluator for postfix notation.	BTL6	Create
17	Construct a circular queue for real-time task scheduling with enqueue and dequeue operations.	BTL6	Create

UNIT IV NON-LINEAR DATA STRUCTURES			
Trees – Binary Trees – Tree Traversals – Expression Trees – Binary Search Tree – Hashing – Hash Functions – Separate Chaining – Open Addressing – Linear Probing– Quadratic Probing – Double Hashing – Rehashing.			
PART – A			
Q.No	Questions	BT Level	Competence
1	Define Abstract Data Types (ADT).	BTL1	Remember
2	What is a tree in data structures?	BTL1	Remember
3	Define a binary tree.	BTL1	Remember
4	Mention the concept of a full binary tree.	BTL1	Remember
5	What is a complete binary tree?	BTL1	Remember

6	Define a binary search tree (BST).	BTL1	Understand
7	State an expression tree.	BTL1	Remember
8	How does a binary search tree differ from a binary tree?	BTL2	Understand
9	How does an expression tree represent arithmetic expressions?	BTL2	Understand
10	What is a hash table?	BTL1	Remember
11	Define a hash function.	BTL1	Remember
12	What is meant by collision in hashing?	BTL1	Remember
13	What is separate chaining in hashing?	BTL1	Remember
14	Define open addressing.	BTL1	Remember
15	State linear probing.	BTL1	Remember
16	What is quadratic probing?	BTL1	Remember
17	Define double hashing.	BTL1	Remember
18	What are the advantages of separate chaining over open addressing?	BTL2	Understand
19	What is the purpose of rehashing in a hash table?	BTL2	Understand
20	Compare the space complexities of separate chaining and open addressing.	BTL2	Understand
21	Construct an expression tree for the expression: $(3 + 5) * (2 - 8)$.	BTL2	Understand
22	Compare the performance of separate chaining and open addressing in hashing.	BTL2	Understand
23	Determine load factor in hashing?	BTL1	Remember
24	State rehashing in hashing.	BTL1	Remember

PART-B

1	What is a Tree in Data Structures? Explain its types.	BTL3	Apply
2	Describe the different tree traversal methods (inorder, preorder, postorder) using a sample tree.	BTL3	Apply
3	Explain the different types of binary trees and their characteristics.	BTL3	Apply
4	Explain how an expression tree represents arithmetic expressions.	BTL3	Apply
5	What is an Expression Tree? Build one for the expression $(a + b) * (c - d)$.	BTL3	Apply
6	Explain the three cases of node deletion in a Binary Search Tree.	BTL3	Apply
7	What is a binary tree? List and define the different types of binary trees.	BTL4	Analyze
8	How are nodes inserted in a Binary Search Tree? Show with example [10, 5, 15, 3, 7].	BTL4	Analyze
9	Construct a binary search tree (BST) using the values [50, 30, 70, 20, 40, 60, 80]. Show the structure.	BTL4	Analyze
10	Explain in detail about separate chaining with example.	BTL4	Analyze
11	Explain collision resolution techniques.	BTL4	Analyze
12	i) Explain the common collision resolution strategies in open address hashing. ii) Describe the different hashing functions with an example.	BTL3	Apply

13	Given input {4371,1323,6173,4199,4344,9679,1989} and a hash function $h(X) = X \pmod{10}$. Show the resulting 1. Separate chaining table 2. Open addressing hash table using linear probing 3. Open addressing hash table using Quadratic probing 4. Open addressing hash table with second hash function $h_2(X) = 7 - (X \pmod{7})$	BTL5	Evaluate
14	Explain Open Addressing in detail.	BTL4	Analyze
15	Using the modulo method, create a hash table of size 7 for keys: 10, 20, 5, 12, 14. Resolve collisions using linear probing.	BTL5	Evaluate
16	Consider a hash table with 9 slots. The hash function is $h(k) = k \pmod{9}$. The following keys are inserted in the order 15, 38, 8, 5, 20, 33, 14, 30. Draw the contents of the hash table when the collisions are resolved by i) Chaining ii) Linear Probing iii) Double hashing. The second hash function $h_2(x) = 7 - (x \pmod{7})$.	BTL6	Create
17	Insert the keys 23, 43, 13, 27 into a hash table using quadratic probing with table size 7. Show each step.	BTL6	Create

UNIT V SORTING AND SEARCHING TECHNIQUES

Insertion Sort – Quick Sort – Heap Sort – Merge Sort – Linear Search – Binary Search.

PART – A

Q.No	Questions	BT Level	Competence
1	What is meant by sorting?	BTL1	Remember
2	List the different sorting algorithms.	BTL1	Remember
3	State internal sorting.	BTL1	Remember
4	Difference between internal and external sorting.	BTL2	Understand
5	Define insertion sort.	BTL1	Remember
6	Illustrate the concept of quick sort. with suitable examples.	BTL2	Understand
7	State merge sort. Give an example.	BTL1	Remember
8	How does insertion sort work?	BTL2	Understand
9	List any two characteristics of heap sort.	BTL1	Remember
10	What is divide-and-conquer strategy?	BTL2	Understand
11	Compare quick sort and merge sort.	BTL2	Understand
12	What is min-heap?	BTL1	Remember

13	Build a max heap from the array [5, 3, 17, 10, 84].	BTL2	Understand
14	Use binary search to find 15 in [3, 7, 15, 20, 25].	BTL2	Understand
15	Analyze why binary search fails on unsorted data.	BTL2	Understand
16	Define Searching.	BTL1	Remember
17	What is linear search?	BTL1	Remember
18	What is meant by Binary search?	BTL1	Remember
19	Differentiate linear search and binary search.	BTL2	Understand
20	State the best case time complexity of binary search.	BTL1	Remember
21	What is the primary condition for using binary search?	BTL1	Remember
22	Why is binary search faster than linear search in sorted lists?	BTL2	Understand
23	Show the first partition step of quick sort on [8, 3, 7, 6].	BTL2	Understand
24	Apply insertion sort on the array: [4, 3, 2, 10]	BTL2	Understand

PART-B			
1	Define and explain insertion sort with an example.	BTL3	Apply
2	State and explain the algorithm for Quick sort with suitable example.	BTL3	Apply
3	Explain the algorithm to perform Heap Sort with Example.	BTL3	Apply
4	Write the insertion sort algorithm and explain with suitable example. Give its worst case, Average Case and Best case time complexities.	BTL3	Apply
5	Write an algorithm to implement Merge sort with suitable example.	BTL3	Apply
6	Write a routine for insertion sort. Sort the following sequence using insertion sort. 3, 10, 4, 2, 8, 6, 5, 1	BTL3	Apply
7	Describe heap construction in heap sort with an example.	BTL4	Analyze
8	Explain quick sort using a pivot and show how partitioning works.	BTL4	Analyze
9	Perform quick sort on [9, 3, 7, 1, 4, 6] and show all steps.	BTL4	Analyze
10	Use heap sort on [10, 20, 15, 30, 40] and show how the heap changes at each step.	BTL4	Analyze
11	Write an algorithm for binary search with suitable example.	BTL4	Analyze
12	Distinguish between linear search and binary search. State and explain the algorithms for both the search with example.	BTL3	Apply
13	Write and trace the steps of binary search to find 20 in the array [10, 15, 20, 25, 30, 35, 40].	BTL5	Evaluate
14	Implement linear search to find if the number 18 exists in the array [12, 17, 23, 18, 31].	BTL5	Evaluate
15	Perform heap sort on the array [4, 10, 3, 5, 1] and show each step.	BTL4	Analyze
16	Explain the working of insertion sort with a step-by-step example.	BTL3	Apply
17.	Apply quick sort to the array [8, 4, 7, 3, 10, 2] and show each step, including pivot selection and partitioning.	BTL4	Analyze