

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY, CSJM UNIVERSITY, KANPUR**

**Subject Name :CSE-S201(CSE)**

**Semester: 2023-24 (Odd Semester)**

**Year: Second Year (2K22)**

**End Semester Examination**

**Time: 3**

**Maximum marks: 50**

All questions are compulsory

**Section A**

10 marks (10 questions of 1 mark each)

1. What is used in breadth-first search algorithms?
2. How is dynamic memory allocation achieved in C using pointers?
3. Define the time complexity of the bubble sort algorithm.
4. How is a circular queue different from a regular queue?
5. What is the difference between a stack and a queue data structure ?
6. What is the significance of a root node in a tree structure?
7. Define the time complexity of the bubble sort algorithm.?
8. How do you check if an element is present in a linked list?
9. What is the time complexity of inserting an element in a linked list?
10. How do you insert a node at the beginning of a linked list in C?

**Section B**

20 marks (5 questions of 4 marks each)

1. What is a stack ? How do you implement a stack using a array?
2. Implement a function to find the middle element of a singly linked list.
3. Explain the difference between a linear search and a binary search.
4. Write a program to find the largest element in an array using pointers.
5. Implement a function to detect a loop in a linked list.

**Section C**

20 marks (2 question of 10 marks.each question have 2 parts)

1. a) Implement a program that uses pointers to count the number of vowels in a string.  
b) Implement a function to merge two sorted arrays into a single sorted array.
- 2) a)Implement a function to delete a node with a given key in a linked list.  
b)Implement a function to reverse a linked list.



University Institute of Engineering and Technology, CSJM University, Kanpur

Department of Computer Science and Engineering

Mid Semester Examination - Data Structure (CSE-S201)

Semester: 2023-24 (Odd Semester)

Year: 2nd Year (2K23)

Time: 1.5 hours

Maximum Marks: 30

All questions are compulsory.

**\*\*Section A (9 Marks - 9 Questions, 1 Mark Each)\*\***

1. Define a pointer in C and explain its significance in memory manipulation.
2. Explain the concept of an array of pointers. Provide an example code snippet to demonstrate its usage.
3. What is a structure in C? How can you declare and initialize a structure variable?
4. Write a C program to implement a stack using an array. Ensure that it includes push and pop operations.
5. Define a singly linked list and explain how to insert a new node at the beginning of the list.
6. Differentiate between the stack and the queue data structures. Provide examples of situations where each is more suitable.
7. Explain the concept of a doubly linked list and how it differs from a singly linked list.
8. What is dynamic memory allocation in C? How is it different from static memory allocation?
9. Discuss the importance of pointer arithmetic in data structure manipulation.

**\*\*Section B (9 Marks - 3 Questions, 3 Marks Each)\*\***

10. Given an array of integers, write a C program to find the second largest element in the array.
11. Explain the concept of a circular queue. Write a C program to implement a circular queue with enqueue and dequeue operations.
12. Create a C program that uses a structure to represent a book with attributes like title, author, and ISBN. Implement a function to display the details of the book.

**\*\*Section C (12 Marks - 2 Questions, 6 Marks Each, Each Question has Two Parts)\*\***

Question 13 a) Write a C program to reverse an array of integers using pointers

b) Explain how you can find the sum of all elements in an integer array using pointer arithmetic.

Question 14 a) Implement a singly linked list to store information about students (e.g., name, roll number, marks). Create functions to add a student, delete a student, and display the list.

b) Discuss the advantages and disadvantages of using a singly linked list to *manage* student records compared to other data structures.



UNIVERSITY INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Information technology

B.Tech. First Mid semester exam, August 2023(2k22)

Paper code: ~~155723~~ CSE-S201

Paper name: DATA STRUCTURE

Duration: 1.5 hours

Total Marks: 30.

Section A: (9 marks)

1: What is a pointer in C?

- a) A variable that holds the value of an expression
- b) A variable that stores the address of another variable
- c) A data type used to store characters
- d) An operator that performs bitwise operations

2: Which of the following statements is true about arrays?

- a) Arrays can only store integers.
- b) Arrays can have a variable number of elements.
- c) The index of the first element in an array is always 0.
- d) Arrays cannot be passed as arguments to functions.

3: What is the result of the following code?

```
```c
int arr[5] = {1, 2, 3, 4, 5};
int *ptr = arr;
printf("%d", *ptr + 2);
```
```

- a) 3
- b) 4
- c) 5
- d) 6

4: In C, the name of an array is a pointer to its first element.

- a) True
- b) False

5: What is the memory representation of a 2D array with dimensions M x N stored in row-major order?

- a) A contiguous block of M\*N elements
- b) An array of M arrays, each containing N elements
- c) A linked list of M elements, each pointing to an array of N elements
- d) A binary tree with M levels, each containing N nodes

6: What is the time complexity for accessing an element in an array using its index?

- a) O(1)



- b)  $O(\log N)$
- c)  $O(N)$
- d)  $O(N^2)$

7: Which of the following is an example of a dynamic memory allocation function in C?

- a) `malloc()`
- b) `array_alloc()`
- c) `allocate()`
- d) `create()`

8: What does the "sizeof" operator return in C?

- a) The value of the variable
- b) The memory address of the variable
- c) The size of the variable in bytes
- d) The number of elements in the variable

9: How can you achieve dynamic memory allocation for arrays in C?

- a) By declaring the array with a fixed size during compile-time
- b) By using the "new" keyword
- c) By using the "malloc" function
- d) By using the "resize" method

#### Section B: (12 marks)

10. Define time complexity. Explain the significance of analyzing time complexity in algorithm design.[3]

5. For each of the following code snippets, calculate the time complexity in terms of "Big O" notation:[3]

- a) `for (int i = 0; i < n; i++) { ... }`
- b) `for (int i = 0; i < n; i *= 2) { ... }`

11. Explain the bubble sort algorithm. Provide a step-by-step example of sorting an array using the bubble sort technique.[3]

12. Describe the insertion sort algorithm. Illustrate its working using a sample array and the step-by-step sorting process.[3]

#### Section C: (9 marks)

13. What is the quick sort algorithm? How does it work? Provide the partitioning steps for an array during the quick sort process.[3]

14. Compare and contrast bubble sort, insertion sort, and quick sort in terms of their average-case time complexity and best-case scenario.[3]

15. Implement the insertion sort algorithm in the C programming language. Provide a code snippet that sorts an array of integers in ascending order.[3]