

Mr. Umesh Thoriya | A.Y.: 2025-26 | Sem.: 4 | Course: 2304CS411 - Data Structure | Division: BCA/BSIT - 4A

Sr	Planning Title	Planning Description
1	Practice decision making statements in C.	1. WAP to find out the largest number from the given two numbers. [A] 2. WAP to find out the largest number from the given three numbers. [A] 3. WAP to perform Addition, Subtraction, Multiplication and Division of 2 numbers as per user's choice. [B] 4. WAP to read marks of five subjects. Calculate percentage and print class accordingly. Fail if percentage is below 35, Pass Class if percentage between 35 to 45, Second Class if percentage is between 45 to 60, First Class if percentage is between 60 to 70, and Distinctions if percentage more than 70. [B]
2	Practice while and do while loop in C.	5. WAP to print 1 to 10 numbers. [A] 6. WAP to print 1 to N numbers. [A] 7. WAP to print odd numbers between 1 to N. [B] 8. WAP to print numbers between two given numbers which is divisible by 2 but not divisible by 4. [B] 9. WAP to print sum of series $1 + 4 + 9 + 16 + 25 + 36 + \dots + n$. [C] 10. WAP to print sum of series $1 - 2 + 3 - 4 + 5 - 6 + 7 - \dots + n$. [C] 11. WAP to print sum of series $1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$. [C]
3	Practice for loop in C.	Practice for loop in C. 12. WAP to print 1 to 10 numbers. [A] 13. WAP to print 1 to n numbers. [A] 14. WAP to print even numbers between 1 to n. [B] 15. WAP to print to find the factorial of a given number. [B] 16. WAP to print the reverse of a given number. [B] 17. WAP to print the following pattern [B] 1 2 3 4 5 6 7 8 9 10 18. WAP to print the following pattern [B] * * * * * * * * * * * * * * *
4	Use array in C.	19. WAP to read n numbers in an array and print them. [A] 20. WAP to read n numbers in an array and print them in reverse order. [A] 21. WAP to find Maximum, Minimum, Sum, and Average of given numbers in an array. [B] 22. WAP to count numbers higher than average of an array. [C]
5	Use structure in C.	23. WAP to create structure of book with book title, author name, publication, and price. Read data of n books and display them. [A] 24. WAP to create structure for employee details (ID, Name, Designation, Salary) and display records of all employees. [A] 25. WAP to read data of students in array of structure and print it. [B] 26. WAP to calculate and display average marks of students using structure. [C]
6	Insertion and Deletion in an Array	27. WAP to insert an element at the Beginning of an Array [A] 28. WAP to insert an element at the End of an Array [A] 29. WAP to delete an element at the Beginning of an Array [B] 30. WAP to delete an element at the end of an Array [B] 31. WAP to insert an element at the Beginning of an Array [C]
7	Implement PUSH POP and Display operations on Stack.	32. Perform following operations on a stack (Consider current stack is empty and stack size is 5) [A]: a. Push: 15, 75, 32 b. Pop c. Push: 14, 16 d. Display e. Pop f. Push: 28, 39, 49 g. Display 33. WAP to implement following operation on stack: PUSH, POP, and Display. [A]

Mr. Umesh Thoriya | A.Y.: 2025-26 | Sem.: 4 | Course: 2304CS411 - Data Structure | Division: BCA/BSIT - 4A

Sr	Planning Title	Planning Description
8	Implement PEEP and CHANGE operations on Stack.	34. Perform following operations on a stack (Consider current stack is S[15, 75, 14, 28, __] and stack size is 5) [A]: a. Change 4th element from top to 46. b. Peep: 3rd element from top. c. Change 2nd element from top to 95. d. Peep: 2nd element from top. 35. WAP to implement following operation on stack: PEEP and CHANGE. [A]
9	Practice conversion of infix notations to postfix notations.	36. Convert following expression to postfix notations [A]: a. $A - B / (C * D ^ E)$ b. $A + B ^ C ^ D - E * F / G$ c. $(A + B * C / D - E + F / G / (H + I))$ d. $(A + B) * C + D / (B + A * C) + D$ e. $A + B - C * D / E + F \$ G / (I + J)$ f. $(a + b ^ c ^ d) * (e + f / d)$
10	Practice conversion of infix notations to prefix notations.	37. Convert following expression to prefix notations [A]: a. $A - B / (C * D ^ E)$ b. $A + B ^ C ^ D - E * F / G$ c. $(A + B * C / D - E + F / G / (H + I))$ d. $(A + B) * C + D / (B + A * C) + D$ e. $A + B - C * D / E + F \$ G / (I + J)$ f. $(a + b ^ c ^ d) * (e + f / d)$
11	Practice evaluation of postfix and prefix notations.	38. Evaluate following expression [A]: a. $5\ 6\ 2\ +\ *\ 1\ 2\ 4\ /\ -\ +$ b. $A\ B\ +\ C\ D\ /\ * \ G\ H\ * \ +$ (where A=2,B=4,C=6,D=3,G=8,H=7) c. $5\ 4\ 6\ +\ * \ 4\ 9\ 3\ /\ +\ *$ d. $+ \ 8\ * \ 3\ 7$ e. $- \ + \ 7\ /\ 4\ 2\ 5$ f. $- \ + \ 4\ * \ 2\ 6\ + \ 4\ 3$
12	Implement simple queue.	39. Perform following operations on a simple Queue (Consider current Queue is empty and queue size is 5) [A]: a. Enqueue: 15, 75, 32 b. Dequeue c. Enqueue: 14, 16 d. Display e. Dequeue f. Enqueue: 28, 39, 46 g. Display 40. WAP to implement the following operation on queue: Enqueue, Dequeue, and Display. [A]

Mr. Umesh Thoriya | A.Y.: 2025-26 | Sem.: 4 | Course: 2304CS411 - Data Structure | Division: BCA/BSIT - 4A

Sr	Planning Title	Planning Description
13	Implement circular queue.	<p>41. Perform following operations on a circular Queue (Consider current Queue is empty and queue size is 5) [A]:</p> <ol style="list-style-type: none"> Enqueue: 15, 75, 32 Dequeue Enqueue: 14, 16 Display Dequeue Enqueue: 28, 39, 46 Display <p>42. WAP to implement the following operation on circular queue: Enqueue, Dequeue, and Display. [A]</p> <p>43. Perform the following operations on a Circular Queue (Queue size = 7) [B]: Consider the current Circular Queue is empty. Perform the operations step by step and show the changes in front and rear pointers after each operation.</p> <ol style="list-style-type: none"> Enqueue: 12, 24, 36, 48, 60 Dequeue two elements Enqueue: 72, 84, 96 Display all elements Enqueue: 108 (Check overflow condition) Dequeue three elements Display the updated Circular Queue
14	Implement Double Ended Queue.	<p>44. Perform the following operations on a Double Ended Queue (Consider current Deque is empty and deque size is 5) [A]:</p> <ol style="list-style-type: none"> Insert Rear: 20, 45, 67 Delete Front Insert Front: 12 Insert Rear: 89 Display Delete Rear Insert Front: 34, 56 Display <p>45. WAP to implement following operation on double ended queue: Insert at front end, Delete from rear, and Display. [A]</p>
15	Revision of the concept of Pointer & Dynamic Memory Allocation using C language	<p>46. WAP to allocate and de-allocate memory for int, char and float variable at runtime. [A]</p> <p>47. WAP to get and print the array elements using Pointer. [A]</p> <p>48. WAP to calculate the sum of n numbers using Pointer. [A]</p> <p>49. WAP to find the largest element in the array using Pointer. [B]</p> <p>50. WAP to define a C structure named Student (roll_no, name, branch and batch_no) and also to access the structure members using Pointer. [C]</p>
16	Implement insert and display operations on linked list.	<p>51. Perform following operations on a link list (Consider current link list is empty) [A]:</p> <ol style="list-style-type: none"> Insert first: 15, 75, 32 Display Insert last: 78, 37, 28 Display <p>52. WAP to implement following operation on link list: Insert at first, Insert at last, and Display. [A]</p> <p>53. WAP to implement following operation on link list: Insert at specified location. [A]</p>

Mr. Umesh Thoriya | A.Y.: 2025-26 | Sem.: 4 | Course: 2304CS411 - Data Structure | Division: BCA/BSIT - 4A

Sr	Planning Title	Planning Description
17	Implement delete operations on linked list.	54. Perform following operations on a link list (Consider current link list is: Start157532783728) [A]: a. Delete first b. Display c. Delete first d. Delete first e. Delete first f. Display 55. WAP to implement following operation on link list: Delete from first, Delete from last. [A] 56. WAP to implement following operation on link list: Delete from specified position. [A]
18	Implement search operation on linked list.	57. Perform following operations on a link list (Consider current link list is: Start157532783728) [A]: a. Search (75) b. Search (94) c. Search (78) 58. WAP to implement following operation on link list: Search node from link list. [A]
19	Implement count node operation on linked list.	59. Count the number of nodes in a link list (Consider current link list is: Start157532783728). [A] 60. WAP to implement the following operation on link list: Count number of node in a link list. [A]
20	Practice conversion of general tree to binary tree.	61. Convert following general tree to binary tree [A]:
21	Practice insertion and deletion operation on binary search tree.	62. Insert and Delete node from following binary tree as per instructions. [A] a. Tree 1: Insert (21), Insert (28), Insert (14), Insert (32), Insert (25), Insert (18), Insert (11), Insert (30), Insert (19), Insert (15) b. Tree 2: Insert (80), Insert (88), Insert (64), Insert (52), Insert (85), Insert (38), Insert (19), Insert (22), Insert (34), Insert (92) c. Tree 3: Insert (74), Insert (28), Insert (96), Insert (83), Insert (75), Insert (35), Insert (48), Insert (31), Insert (16), Insert (4) d. Tree 4: Insert (59), Insert (38), Insert (45), Insert (76), Insert (66), Insert (19), Insert (26), Insert (55), Insert (72), Insert (49), Delete (26), Delete (19), Delete (49), Delete (72) e. Tree 5: Insert (57), Insert (34), Insert (38), Insert (66), Insert (86), Insert (99), Insert (36), Insert (25), Insert (42), Insert (40), Delete (42), Delete (86), Delete (66) f. Tree 6: Insert (56), Insert (49), Insert (65), Insert (68), Insert (38), Insert (50), Insert (63), Insert (64), Insert (76), Insert (66), Delete (65), Delete (68), Delete (56)
22	Practice tree traversal.	63. Find out in order, Pre order and Post order traversal of tree given below [A]:
23	Practice tree traversal.	64. Find out in order, Pre order and Post order traversal of tree given below [A]:
24	Implement linear search algorithm.	65. Search element 64 in Array A [75, 2, 34, 19, 73, 64, 85, 26, 15, 10, 28] using linear search algorithm. [A] 66. WAP to implement linear search algorithm. [A]
25	Implement binary search algorithm.	67. Search element 44 in Array A [5, 12, 34, 44, 65, 68, 70, 76, 85, 90, 128] using binary search algorithm [A] 68. WAP to implement binary search algorithm. [A]
26	Implement bubble sort algorithm.	69. Sort Array A [75, 2, 34, 19, 73, 64, 85, 26, 15, 10, 28] using bubble sort algorithm. [A] 70. WAP to implement bubble sort algorithm. [A]
27	Implement selection sort algorithm.	71. Sort Array A [75, 2, 34, 19, 73, 64, 85, 26, 15, 10, 28] using selection sort algorithm. [A] 72. WAP to implement selection sort algorithm. [A]
28	Implement insertion sort algorithm.	73. Sort Array A [75, 2, 34, 19, 73, 64, 85, 26, 15, 10, 28] using insertion sort algorithm. [A] 74. WAP to implement insertion sort algorithm. [A]

Mr. Umesh Thoriya | A.Y.: 2025-26 | Sem.: 4 | Course: 2304CS411 - Data Structure | Division: BCA/BSIT - 4A

Sr	Planning Title	Planning Description
29	Implement merge sort algorithm.	75. Sort Array A [75, 2, 34, 19, 73, 64, 85, 26, 15, 10, 28] using merge sort algorithm. [A] 76. WAP to implement merge sort algorithm. [A]
30	Solving real life problem using sorting techniques	77. Write a program to sort an array containing 0s, 1s, and 2s (representing red, white, and blue colors respectively) in such a way that all objects of the same color are adjacent, and colors are arranged in the order red (0), white (1), and blue (2). [A] Input: nums = [2,0,2,1,1,0] Output: [0,0,1,1,2,2]