

U.G. 3rd Semester Examination - 2020

COMPUTER SCIENCE

[HONOURS]

Course Code : COM.SC-H-CC-P-305

(Data Structures Lab)

[PRACTICAL]

Full Marks : 75

Time : 6 Hours

The figures in the right-hand margin indicate marks.

Marks Distribution :

Experiment : 60 Marks

Viva-voce : 10 Marks

Lab Notebook : 05 Marks

Answer any **two** to be allotted on lottery basis.

30×2=60

1. Write a program to count and display the number of occurrences of each element of an array without displaying the duplicate ones.
2. Write a program to create an array. Make options to insert and delete numbers at user-given positions. Display the array after each operation.
3. Write a program to create an array of size fifteen and insert at least three elements three times. Then delete all the duplicate elements of the array. Display the array after each operation.

[Turn over]

4. Write a menu driven program to perform the following operations in matrix: (i) addition (ii) subtraction (iii) multiplication.
5. Write a program to calculate inverse of a matrix.
6. Write a program to reverse the order of elements of a stack.
7. Write a program to concatenate two doubly linked list.
8. Write a program to display diagonal matrix using one dimensional array.
9. Write a menu driven program to implement stack.
10. Write a program to implement a linked list, with the option to insert a node into it at the user given position.
11. Write a program to implement a linked list, with the option to delete a node from it at the user given position.
12. Write a program to search an element in a linked list.
13. Write a program to reverse a linked list.
14. Write a program to implement in-place insertion sort.
15. Write a menu driven program to implement a circular queue using circular linked list.
16. Write a program to implement iterative binary search.
17. Write a program to implement binary search using recursion.

18. Write a program to implement merge sort using recursion.
19. Write a program to convert an infix expression to post-fix expression. The program should support both parenthesized and non-parenthesized expression.
20. Write a program to implement quick sort.
21. Write a menu driven program to create a BST and add option to insert node.
22. Write a menu driven program to create a BST and add option to delete node.
23. Write a program to search an element in a BST.
24. Write a menu driven program to create a max heap. Add options to insert and delete nodes.
25. Write a program to implement heap sort.
26. Write a program to implement preorder traversal of a binary tree having ten nodes.
27. Write a program to implement post-order traversal of a binary tree having ten nodes.
28. Write a program to merge and sort two unsorted arrays.
29. Write a program to implement hashing on a hash table of size ten. Resolve the collisions by linear probing technique.
30. Write a program to implement multi-level indexing.
