TABLE OF CONTENTS

|  |  |  |  |
| --- | --- | --- | --- |
| Ass. No | SL No. | Problem Statement | Page |
| Assignment 1 | 1. | Write a program to find the maximum element in an array. | 1 |
| 2. | Implement a function to revere an array in place. | 2 |
| 3. | Implement a function to reverse an array in place. | 3 |
| 4. | Write an algorithm to rotate an array given number of positions. | 5 |
| 5. | Implement an algorithm to find the missing number in an array of integers from 1 to N. | 7 |
| Assignment 2 | 6. | Write a function to remove duplicates from sorted array. | 8 |
| 7. | Implement an algorithm to find the majority element in an array. | 10 |
| 8. | Create a program to find the largest subarray with a sum less than or equal to a given value. | 11 |
| 9. | Write a function to find the contiguous subarray with the largest sum (Kadane’s algorithm). | 13 |
| 10. | Implement an algorithm to search for an element in a sorted and rotated array (rotated binary search). | 15 |
| Assignment 3 | 11. | Create a single linked list and write a function to reverse it. | 17 |
| 12. | Implement a program to detect if a linked list has a cycle. | 21 |
| 13. | Write a function to merge two sorted linked list into a single linked list. | 25 |
| 14. | Implement an algorithm to find the Nth node from the end of a linked list. | 30 |
| 15. | Create a program to delete a node with a given value from a linked list. | 34 |
| 16. | Write a function to check if 2 linked list intersect and if they do find intersection node. | 39 |
| 17. | Implement a function to add 2 numbers represented by linked lists (e.g., 342 + 465 = 807). | 43 |
| Assignment 4 | 18. | Create a function to sort n array using bubble sort. | 48 |
| 19. | Write a program to perform the quick sort. | 50 |
| 20. | Implement the heapsort algorithm. | 52 |
| 21. | Implement the merge algorithm. | 55 |
| 22. | Implement the radix sort algorithm for integers. | 58 |
| Assignment 5 | 23. | Implement a binary search tree and write functions for insertion & deletion. | 61 |
| 24. | Write a program to find the height of a binary tree. | 66 |
| 25. | Implement an algorithm to check if a binary tree is a binary search tree(BST). | 70 |
| 26. | Create a function to find the lowest common ancestor (LCA) Of two nodes in a binary tree. | 73 |