Part - C

PROGRAMMING ASSIGNMENTS

1. Insert an element at any position of an array
2. Delete an element at any position of an array
3. Interchange each 2 alternate element of an array.

   I/P - 10 , 20 , 30 , 40 , 50

   O/P - 20 , 10 , 40 , 30 , 50
4. Sort the elements of an array using bubble sort.
5. Sort the elements of an array using selection sort.
6. Sort the elements of an array using insertion sort.
7. Sort the elements of an array using quick sort.
8. Sort the elements of an array using sequential sort.
9. Sort the elements of an array using merge sort.
10. Sort the elements of an array using heap sort.
11. Sort the elements of an array using Topological sort.
12. Sort the elements of an array using radix sort.
13. Sort the elements of an array according to their frequencies

    2,2,2,6,6,6,6,9,9,9,5,5,1,1,1,1,1

    1,1,1,1,1,6,6,6,6,9,9,9,2,2,2,5,5
14. There is an array with an input {0,1,2,3,5,6,9} . Write a program to show the combinations for which the sum is 9. Ex - {3,6} , {0,9} ... etc
15. Search an element in an array using binary search.

16. Search an element in an array using linear probing.
17. Search an element in an array using Chaining mechanism
18. Remove the duplicate elements of an array.
19. Linear search using an array
20. Write a program to separate the elements of an array as follows –

    I/P - 1 , 0 , 1 , 0 , 0 , 0 , 1 , 1 , 0

    O/P - 0 , 0 , 0 , 0 , 0 , 1 , 1 , 1 , 1

21 Write a program to convert a string in following format –

    I/P - " aaaaabbbccccdd "

    O/P - a5b3c4d2

22. Write a program to convert a string in the following format

    I/P - a5b3c4d2

    O/P - " aaaaabbbccccdd "

23. Program for matrix multiplication.
24. Program for matrix addition.
25. Write a program to add all upper triangular elements of a matrix
26. Program to transpose a matrix.
27. Program to transpose a sparse matrix.
28. Program to add between two sparse matrices.
29. Multiply two matrices if matrix is provided using array of pointers.
30. Write a program to check whether a string is balanced or not.

    I/P - "{([([[]]])} "

31. Find a sub string in a string.

32. There are two array exist , such as x and y. All elements of array 'x' is present in array 'y' with extra elements. Write a program to find the extra elements present in array 'y'.
33. Find the second largest element in a linked list.
34. Find the nth element from the last of linked list
35. Remove the duplicate nodes from a linked list
36. Insert a node at any position in a double linked list
37. Remove a node from any position in a double linked list
38. Interchange each 2 alternate nodes of a linked list
39. Implement linear search in a linked list
40. Implement binary search in a linked list.
41. Reverse a linked list
42. Add between any two numbers using link list
43. Remove the duplicate nodes from a link list
44. Delete a node from a linked list without knowing its previous node address.
45. Efficient way to find the middle of link list.
46. Create a polynomial using a linked list.
47. Add between two polynomial using a linked list.
48. Find the second largest node in a link list
49. Modify the linked list such that all even numbers appear before all the odd numbers in the modified linked list.
50. Write a program to insert and delete element at any position of a double link list.
51. Write a C program to push and pop elements in a stack.
52. Write a C program to sort elements in a stack.
53. Write a C program to convert a number into binary using stack.

54. Write a C program to convert an infix expression to postfix expression.
55. Write a C program to convert an infix expression to prefix expression.
56. Write a C program to evaluate prefix expression.
57. Write a C program to evaluate postfix expression.
58. Write a C Program to find the smallest element in the stack.
59. Implement a queue using two stack.
60. Write a C Program to Check whether the following sequence is balanced or not using stack – "{[()]}".
61. Perform the enque, deque and traversal operations in a circular queue using array.
62. Perform the enque, deque and traversal operations in a priority queue using linked list.
63. Write a c program to create an input restricted queue.
64. Write a c program to implement priority queue using two stack.
65. Write a C program for DFS traversal of Graph.
66. Write a C program for BFS traversal of Graph.
67. Write a C Program to find the shortest path using Warshal's algorithm.
68. Write a C Program to find the shortest path using Dijkstra's algorithm.
69. Find the minimum spanning tree using Kruskal's alogorithm
70. Find the minimum spanning tree using Prim's alogorithm
71. Write a program to check the graph is cyclic or acyclic
72. Write a program to find path matrix of graph

74. Write a C program to traverse a tree in label order
75. Write a C program to find the height of the tree
76. Write a C program to create a tree using Binary search Tree and traverse binary search tree
77. Create a tree using Heap
78. Write a program for Threaded Binary Tree
79. Write a program to count number of nodes present in a tree
80. Check whether two trees are structurally identical or not.

Answers

1. Insert an element at any position of an array.

```
main()
{
      int x[5]={10,20,30,40,50};
      int n,i,pos;
      printf("Enter position:");
      scanf("%d",&pos);
      printf("Enter data:");
      scanf("%d",&n);
      for(i=4;i>pos;i--)
      {
            x[i]=x[i-1];
      }
      x[pos]=n;

      for(i=0;i<5;i++)
            printf("%d ",x[i]);
      }
```