**Datastructures Lab Task II/IV B.Tech**

Topic: Applications of LinkedStacks&Queues

Batch-1: 01-February-2022 @ 9.30 to 12

Batch-2: 03- February -2022 @ 8.40 to 11.10

**Task-1**

In a class room we have discussed the algorithms for different traversal techniques like (i) Inorder (ii) preorder and (iii) postorder. Based on the same discussion implement a solution. Use the following LOC and complete the implimenation part. Also run all the possible testcases.

1. struct node
2. {
3. int num;
4. struct node \*left;
5. struct node \*right;
6. };
7. void main()
8. {
9. struct node \*v1,\*v2,\*v3,\*v4,\*v5,\*v6,\*v7;
10. v1=create();
11. v2=create();
12. -----------------
13. -----------------
14. -----------------
15. -----------------
16. -----------------
17. v1->left=v2;
18. v1->right=v3;
19. v2->left=v4;
20. v2->right=v5;
21. v3->left=v6;
22. v3->right=v7;
23. printf("\nInorder:");
24. inorder(v1);
25. printf("\nPreorder:");
26. preorder(v1);
27. printf("\nPostorder:");
28. postorder(v1);
29. getch();
30. }
31. struct node \*create()
32. {
33. struct node \*temp;
34. temp=(struct node \*)malloc(sizeof(struct node));
35. temp->left=NULL;
36. temp->right=NULL;
37. printf("Enter value:");
38. scanf("%d",&temp->num);
39. return temp;
40. }
41. void inorder(struct node \*T)
42. {
43. if(T!=NULL)
44. {
45. inorder(T->left);
46. inorder(T->right);
47. }
48. }
49. void preorder(struct node \*T)
50. {
51. }
52. }
53. void postorder(struct node \*T)
54. {
55. if(T!=NULL)
56. {
57. postorder(T->left);
58. postorder(T->right);
59. printf("%d",T->num);
60. }
61. }

**Task-2**

In a class room we have discussed the algorithms for infix to postfix conversion. Based on the same discussion implement a solution. Also run all the possible testcases.

**Testcase-1**

**Infix Expression : 3+4\*5/6**

**Postfix Expression : 3 4 5 \* 6 / +**

**Testcase-2**

**Infix Expression :** A+(B\*C-(D/E^F)\*G)\*H

**Postfix Expression :** ABC\*DEF^/G\*-H\*+