Shreshtha Aggarwal
1BM19CS155

Date
Page 67

21|12|20 Lab - 10

Q> WAP for Binary tree.

```
# include < stdio.h >
# include < stdlib.h >
# include < string.h >
struct node {
        int info;
        struct node *llink, *rlink; };
typedef struct node *Node;
Node getnode () {
        Node x;
        x = (Node) malloc (sizeof ( struct node));
        if (x == NULL) {
            printf (" Memory not available\n");
            exit (0); }
        return x; }
void freenode (Node x) { free (x); }
Node insert ( int item, Node root ) {
        Node temp, cur, prev;
        char direction [10];
        int i;
        temp = getnode ();
        temp -> info = item;
        temp -> llink = NULL;
        temp -> rlink = NULL;
        if ( root == NULL) { return temp; }
```

```
printf(" give dir" to insert : ");
scanf(" %s", direction);
prev = NULL;
cur = root;
for (i=0; i< strlen(direction) &&
        cur != NULL; i++) { prev=cur;
    if (direction [i] == 'l' || direction
        [i] == 'L') { cur = cur -> llink; }
    else { cur = cur -> rlink; } }
if (cur != NULL || i != strlen (direction)){
    printf ("Insertion not possible \n");
    freenode ( temp);
    return (root); }
if (cur == NULL) {
    if (direction [i-1] == 'l' ||
        direction [i-1] == 'L') {
        prev -> llink = temp; }
    else { prev -> rlink = temp; }}
return (root); }
void preorder (Node root) {
    if (root != NULL) {
        printf (" %d -> \n", root -> info);
        preorder (root -> llink);
        preorder (root -> rlink); }}
void inorder (Node root) {
      if ( root != NULL ) {
            inorder ( root -> llink );
            printf (" %d -> ", root -> info);
            inorder ( root -> rlink ); } }
void postorder ( Node root ) {
      if ( root != NULL ) {
            postorder ( root -> llink );
            postorder ( root -> rlink );
            printf (" %d -> ", root -> info)}}
void display ( Node root, int i) {
      int j ;
      if ( root != NULL) {
         display ( root -> rlink , i+1 );
         for ( j=1; j<=i; j++ )
               printf (" ");
         printf ("%d \n", root -> info);
         display ( root -> llink , i+1 ) ;}}
int main () {
      Node root = NULL;
      int choice, item;
      for (;;) {
         printf ("1 Insert \n2 Preorder \n3.
            Inorder \n 4 Postorder \n 5.
            Display \n 6. Exit \n");
         printf (" Enter choice : ");
```

```
    scanf("%d", &choice);
    switch(choice){
        case1: printf("Enter item : ");
            scanf("%d", &item);
            root = insert (item, root); break;
        case 2: printf("Preorder traversal : ");
            preorder(root);
            break;
        case3: printf("Inorder traversal : ");
            inorder(root);
            break;
        case 4 : printf(" Postorder traversal:");
            postorder(root);
            break;
        case 5: display (root, 1);
            break;
        case 6: exit (0);
        default: printf("Enter proper value\n");
            break; } }
    return 0;
}
```

O/P ->

```
1. Insert  2 Preorder 3 Inorder 4. Postorder
5. Display 6. Exit
Enter choice : 1
Enter item : 10
```