LAB -10

HEMANG SINGH

```
1) # ~~void~~ include < stdio.h>
   # include<stdlib.h>
   struct node {
         int info;
         struct node  * rlink , llink;

   };

   typedef struct node*  NODE;
   NODE getnode () {
        NODE x;
        x = ( NODE) malloc( sizeof (struct node));
        if( x == NULL) {
           printf(" Memory full \n");
           exit(0);
        }
        return x;
   }
   void freenode ( NODE x) {
        free(x);
   }
   NODE insert ( NODE root, int item) {
        NODE temp, cur, prev;
```

```
temp = getnode();
temp -> rlink = NULL;
temp -> llink = NULL;
temp -> info = item;

if( root == NULL)
        return temp;

  prev = NULL;
  cur = root;
  while( cur! = NULL) {

      prev = cur;
      cur = (item < cur->info) ?
                      cur -> llink : cur->rlink;

 }
if (item < prev -> info)

      prev -> llink = temp;

      else {
          prev -> rlink = temp;

      return root;

 }
Void display (NODE root, int i) {
        int j;
        if( root ! = NULL)
```

```
    {
      display( root -> rlink , i+1);
      for( j = 0; j < i ; j++)
            printf("  ");
      printf(" %d\n", root -> info );
      display( root -> llink , i+1);

    }
 }
 Void preorder(NODE root){
      if( root != NULL){
            printf(" %d \n", root -> info);
            preorder( root -> rlink);
            preorder( root -> llink);

      }
 }
 void inorder( NODE root){
      if( root != NULL){
            inorder( root -> llink);
            printf(" %d \n", root -> info);
            inorder( root -> rlink);
      }
 }
```

```
int main() {
    int item, choice;
    NODE root = NULL;
    for(;;) {
    printf(" \n1. Insert\n 2. Display \n3. Preorde
        \n4. Postorder \n5. Inorder \n6. exit
          \n");

    printf(" Enter the choice \n");
    scanf(" %d", & choice);
     switch( choice) {

      case 1: printf(" Enter the item \n");
              scanf(" %d", & item);
              root = insert (root, item);
              break;

      case 2: display (root, 0);
              break;

      case 3: preorder( root);
              break;

      case 4: postorder (root);
              break;

      case 5: inorder (root);
              break;

     default: ~~exit~~(0);
              break;
} } }
```