```c
//Lab Program 10
//Binary search tree

#include <stdio.h>
#include <stdlib.h>

typedef struct BST{
    int data;
    struct BST *left,*right;
}node;

node *create(){
    node *newnode;
    int value;
    printf("Enter value: ");
    scanf("%d",&value);
    newnode = (node*)malloc(sizeof(node));
    newnode->data = value;
    newnode->left = NULL;
    newnode->right = NULL;
    return newnode;
}
void insert(node *root,node *temp){
    if(temp->data < root->data){
        if(root->left == NULL)
            root->left = temp;
        else{
            insert(root->left,temp);
        }
    }
    if(temp->data > root->data){
        if(root->right == NULL)
            root->right = temp;
        else{
            insert(root->right,temp);
        }
    }
}
void inorder(node *root){
    if(root != NULL){
        inorder(root->left);
        printf("%d\t",root->data);
        inorder(root->right);
    }
```

```
41          inorder(root->left);
42          printf("%d\t",root->data);
43          inorder(root->right);
44      }
45  }
46  void preorder(node *root){
47      if(root != NULL){
48          printf("%d\t",root->data);
49          preorder(root->left);
50          preorder(root->right);
51      }
52  }
53  void postorder(node *root){
54      if(root != NULL){
55          postorder(root->left);
56          postorder(root->right);
57          printf("%d\t",root->data);
58      }
59  }
60  //node *minvaluenode(){}
61  //node *deletenode(){}
62  int main(int argc,char **argv){
63      int choice;
64      node *root=NULL,*temp;
65      while(choice != 5){
66          printf("\nEnter choice 1)insert 2)inorder 3)preorder 4)postorder 5)exit : ");
67          scanf("%d",&choice);
68          switch(choice){
69              case 1:temp = create();
70                     if(root == NULL)
71                         root = temp;
72                     else
73                         insert(root,temp);
74                     break;
75              case 2:inorder(root);break;
76              case 3:preorder(root);break;
77              case 4:postorder(root);break;
78              case 5:
79              default:exit(0);
80          }
81      }
82      return 0;
83  }
84
```