Date ______
Page ______

LAB 10.

```
# include <stdio.h>
# include <stdlib.h>
struct btnode{
        int value;
        struct btnode *l;
        struct btnode *r;
}*root = NULL , *temp=NULL, *t2 , *t1 ;
~~[illegible]~~
void insert ();~~{~~
void inorder (struct btnode *t);
void create ();
void search(struct btnode *t);
void preorder (struct btnode *t);
void postorder (struct btnode *t);
int flag = 1;
 void main(){
     int ch;
     printf ("1. Insert\n 2. Inorder\n 3. Preorder\n 4. Postorder\n 5. Exit");
     while (1){
          printf("Enter choice: ");
          scanf("%d", &ch);
          switch(ch){
               case 1: insert (); break;
               case 2: inorder (); break;
               case 3: preorder (); break;
               case 4: postorder(); break
     }         case 5: exit(0);
      }
}
}
```

s.

```
void insert (){
        create();
        if(root == NULL)
            root = temp;
        else
            search(root);
}

void create (){
        int data;
        printf("Enter data ");
        scanf("%d", &data);
        temp= (struct btnode* )malloc (1*sizeof(struct ~~node~~ btnode));
        temp→value = data;
        temp→l = temp→r =NULL;
}
void search (struct btnode *t){
    if((temp→ value >t→value) && (t→r != NULL))
        search(t→r);
    else if ((temp→ value >t→value) && (t→r == NULL))
        t →r =temp;
    else if((temp→value < t→ value) && (t→l != NULL))
        search(t→l);
    else if ((temp→value < t→value ) && (t→l =NULL))
        t→ l = temp;
}
void inorder(struct btnode *t){
    if(root == NULL){
        printf("Empty tree");
        return;
    }
```

Date ________
Page ________

```
    if(t->l != NULL)
        inorder(t->l);
    printf("%d->", t->value);
    if(t->r != NULL)
        inorder(t->r);
}
void preorder(struct btnode *t){
    if(root == NULL){
        printf("Empty tree");
        return;
    }
    printf("%d->", t->value);
    if(t->l != NULL)
        preorder(t->l);
    if(t->r != NULL)
        preorder(t->r);
}

void postorder(struct btnode *t){
    if(root == NULL){
        printf("Empty tree");
        return;
    }
    if(t->l != NULL)
        postorder(t->l);
    if(t->r != NULL)
        postorder(t->r);
    printf("%d->", t->value);
}
```