

```

/*
name : Durvesh D. Dhake
section: A
roll no: A-48
batch: A-2
*/
// 10. A Program to Construct a Tree & Perform Insertion, Deletion, Display
// C Program to Construct a Tree & Perform Insertion and Display
#include <stdio.h>
#include <stdlib.h>
struct btnode
{
int value;
struct btnode *l;
struct btnode *r;
}*root = NULL;
// Function Prototype
void printout(struct btnode*);
struct btnode* newnode(int);
void main()
{
root=newnode(50);
root->l=newnode(20);
root->r=newnode(30);
root->l->l=newnode(70);
root->l->r=newnode(80);
root->l->r->r=newnode(60);
root->l->l->l=newnode(10);
root->l->l->r=newnode(40);
printf("tree elements are\n");
printf("\nDISPLAYED IN INORDER\n");
printout(root);
printf("\n");
}
// Create a node
struct btnode* newnode(int value)
{
struct btnode* node = (struct btnode*)malloc(sizeof(struct btnode));
node->value = value;
node->l = NULL;
node->r = NULL;
return(node);
}
// to display the tree in inorder
void printout (struct btnode *tree)
{
if(tree->l)
printout(tree->l);
printf("%d->", tree->value);
if(tree->r)
printout(tree->r);
}
}
/*
output:
tree elements are

```



DISPLAYED IN INORDER

10->70->40->20->80->60->50->30->

\*/

