10) Binary Search Tree Program

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
 struct Node *left;
 int data ;
 struct Node *right;
}*node;
node getnode (int item) {
node temp =(node) malloc (size of (struct node));
 temp -> left = NULL;
temp -> data = item;
 temp -> right = NULL;
 return temp;
}
node insert (node root, int ele) {
if (root = = NULL)
return getnode (ele);
else if (ele < root -> data)
root -> left = insert (root -> left, ele);
else if (ele > root -> data)
 root -> right = insert (rootright, ele);
return root;
}
void inorder (node root) {
if (root = = NULL)
return;
inorder (root -> left);
printf ("%d", root -> data);
inorder (root -> right);
```

```
}
void preorder(node root) {
if (root ==NULL)
return;
printf("%d", root->data);
preorder (root -> left) ;
preorder (root -> right;
}
void postorder(node root) {
if (root ==NULL)
return;
postorder (root -> left);
postorder (root -> right);
printf("%d ", root -> data);
}
int main() {
node root=NULL;
int e, ch=1;
while (ch!=5) {
printf("\n1.Insert\n2.PreOrder Display\n3.InOrder DISPLAY\n
4.PostOrder Display\n~~"~~, 5.Exit\n");
scanf("%d", &ch);
printf("\n");
switch(ch) {
Case 1: printf("element: ");
scanf("%d", &e);
root = insert(root, e); break;
case 2: preorder (root);
break;
case 3 : inorder (root) ; break;
case 4: postorder (root) ; break;
case 5: printf("Exiting.");
exit(1);
default: printf("Wrong Input!");
}
}
}
```