```c
//By Ritwik Chandra Pandey
//On 1st Sept. 2021
//BST Operations - Deletion and Pre-order traversal

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

struct node {
        int data;
        struct node *left, *right;
};
typedef struct node * BSTNODE;
BSTNODE newNodeInBST(int item) {
        BSTNODE temp =  (BSTNODE)malloc(sizeof(struct node));
temp->data = item;
        temp->left = temp->right = NULL;
        return temp;}
void preorderInBST(BSTNODE root) {
        if(root==NULL){
                return;
        }
        printf("%d ",root->data);
preorderInBST(root->left);
        preorderInBST(root->right);}
BSTNODE insertNodeInBST(BSTNODE node, int ele) {
        if (node == NULL) {
                printf("Successfully inserted.\n");
                return newNodeInBST(ele);
        }
        if (ele < node->data)
                node->left  = insertNodeInBST(node->left,ele);
        else if (ele > node->data)
                node->right = insertNodeInBST(node->right,ele);
        else
                printf("Element already exists in BST.\n");
        return node;}
BSTNODE minValueNode(BSTNODE node) {
        while(node->left!=NULL){
                node = node->left;
        }
```

```c
        return node;}
BSTNODE deleteNodeInBST(BSTNODE root, int ele) {
        if(root==NULL){
                printf("Cannot find %d in the binary search tree.\n",ele);
                return root;
        }
        if(ele < (root->data))
            root->left = deleteNodeInBST(root->left,ele);
        else if(ele > (root->data))
            root->right= deleteNodeInBST(root->right,ele);
        else{
                if(root->left == NULL){
                        BSTNODE temp= root->right;
                        printf("Deleted %d from binary search tree.\n",ele);
                        free(root);
                        return temp;
                }
                else if(root->right==NULL){
                        BSTNODE temp = root->left;
                        printf("Deleted %d from binary search tree.\n",ele);
                        free(root);
                        return temp;
                }
                BSTNODE temp = minValueNode(root->right);
                root->data =temp->data;
                temp->data= ele;
                root->right = deleteNodeInBST(root->right,temp->data);
}
        return root;

}




void main() {
        int x, op;
BSTNODE root = NULL;
        while(1)
{       printf("1.Insert 2.Delete 3.Preorder Traversal 4.Exit\n");
```

```c
printf("Enter your option : ");
                scanf("%d", &op);
switch(op) {
    case 1:        printf("Enter an element to be inserted : ");
        scanf("%d", &x);
        root = insertNodeInBST(root,x);
break;
case 2:printf("Enter an element to be deleted : ");
scanf("%d", &x);
                root = deleteNodeInBST(root,x);
break;
        case 3:
if(root == NULL) {printf("Binary Search Tree is empty.\n");
}
else {
printf("Elements of the BST (pre-order traversal): ");
preorderInBST(root);
printf("\n");
                                        }
break;
                        case 4:exit(0);
}
        }
}
```