Both the assignment is included in this prog i am sorry i should have done in a new prog but it just seemed waste of time to write the same input thing twice..pardon me for that but need to save some storage i guess

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
    int data;
    struct node *left;
    struct node *right;
} ab;
ab *root, *temp, *temp1;
ab *createnode(ab *root, int l)
{
    if (root == NULL)
    {
        root = malloc(sizeof(ab));
        root->data = l;
        root->left = root->right = NULL;
        return root;
    }
    if (l > root->data)
    {
        root->right = createnode(root->right, l);
    }
    else if (l < root->data)
    {
        root->left = createnode(root->left, l);
    }
    else
        printf("same data cannot be entered inside a binary search tree or binary tree in general\n");
    return root;
}
void postorder(ab *root)
{
    if (root != NULL)
    {
        postorder(root->left);
        postorder(root->right);
        printf("%d ", root->data);
    }
```

Both the assignment is included in this prog i am sorry i should have done in a new prog but it just seemed waste of time to write the same input thing twice..pardon me for that but need to save some storage i guess

```
    else
        return;
}
void preorder(ab *root)
{
    if (root != NULL)
    {
        printf("%d ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
    else
        return;
}
void inorder(ab *root)
{
    if (root != NULL)
    {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
    else
        return;
}
int getrightmin(ab *root)
{
    temp1 = root;
    while (temp1->left != NULL)
        temp1 = temp1->left;
    return temp1->data;
}
ab *delete (ab *root, int n)
{
    if(root==NULL)
    {
        printf("the element doesnt exist in the BST\n");
        return root;
    }
```

Both the assignment is included in this prog i am sorry i should have done in a new prog but it just seemed waste of time to write the same input thing twice..pardon me for that but need to save some storage i guess

```
    if (n != root->data)
    {
        if (n < root->data)
        {
            root->left = delete (root->left, n);

        return root;
        }

        if (n > root->data)
        {
            root->right = delete (root->right, n);

        return root;
        }
    }
    else
    {
        if (root->left == NULL && root->right == NULL)
        {
            free(root);
            return NULL;
        }
        if (root->left != NULL && root->right == NULL)
        {
            temp = root->left;
            free(root);

            return temp;
        }
        if (root->left == NULL && root->right != NULL)
        {
            temp = root->right;
            free(root);

            return temp;
        }
        if (root->left != NULL && root->right != NULL)
        {
```

Both the assignment is included in this prog i am sorry i should have done in a new prog but it just seemed waste of time to write the same input thing twice..pardon me for that but need to save some storage i guess

```
            temp = root;
            int rightmin = getrightmin(root->right);
            root->data = rightmin;
            root->right = delete (root->right, rightmin);
        }
    }
    return root;
}

int main()
{
    int a;
    printf("enter the number of nodes you want in your binary search
tree\n");
    scanf("%d", &a);
    root = NULL;
    for (int i = 1; i <= a; i++)
    {
        int l;
        printf("enter the data you want to enter in the node\n");
        scanf("%d", &l);
        root = createnode(root, l);
    }
    printf("this is the postorder of the binary search tree-------->\n");
    postorder(root);
    printf("\n");
    printf("this is the preorder of the binary search tree-------->\n");

    preorder(root);
    printf("\n");
    printf("this is the inorder of the binary search tree-------->\n");

    inorder(root);
    int n;
    printf("\nenter the number you want to delete from the node \n");
    scanf("%d", &n);
    root = delete (root, n);
    printf("this is the postorder of the binary search tree after
deletion-------->\n");
```

Both the assignment is included in this prog i am sorry i should have done in a new prog but it just seemed waste of time to write the same input thing twice..pardon me for that but need to save some storage i guess

```
    postorder(root);
    return 0;
}
```