

```

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

typedef struct bst {
    int data;
    struct bst * left;
    struct bst * right;
}bst;

bst * root = NULL;

bst * createnode(int data) {
    bst * newnode = (bst *)malloc(sizeof(bst));
    newnode->data = data;
    newnode->left = NULL;
    newnode->right = NULL;
    return newnode;
}

bst * insert (bst * temp , int data ) {
    if(temp == NULL)
        temp = createnode(data);

    else if (data <= temp->data)
        temp->left = insert(temp->left , data);

    else
        temp->right = insert(temp->right , data);

    return temp;
}

void display(bst * temp , int i) {
    int j;
    if(temp != NULL) {
        display(temp->left , i+1);
        for(j = i ; j<=i ; j++ )
            printf("\t");

        printf("%d \n",temp->data);
        display(temp->right , i+1);
    }
}

void main() {

    printf("Enter the operation to be performed.\n");
    int choice , flag =1, bit;
    while(flag) {
        printf("1.Insert \t 2.Display \t 3.Exit : ");
        scanf("%d", &choice);

        switch(choice) {
            case 1 : printf("Enter the element to bew
inserted : ");
                    scanf("%d",&bit);

```

```

                                root = insert (root , bit);
                                break;
        case 2 : if(root == NULL)
                                printf("Binary search
tree has not been created yet.\n");
                                else
                                        display(root, 1);
                                break;
        case 3 : flag = 0;
                                break;
        default : printf("Invalid input.\n");
                                break;
    }
}

```