Mazni Salwa
1DM19CS201

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

struct node
{
    struct node * lchild;
    int info;
    struct node * rchild;
};

struct node * insert(struct node * ptr, int ikey);
struct node * Min(struct node * ptr);
struct node * Max(struct node * ptr);
void display(struct node * ptr, int level);

int main()
{
    struct node * root = NULL, *ptr;
    int choice, k;

    while (1)
    {
        printf("\n");
        printf(" 1. Insert \n");
        printf(" 2. Display \n");
        printf(" 3. Find min & max \n");
        printf(" 4. Quit \n");
        printf(" \n Enter ur choice : ");
        scanf(" %d", &choice);
        switch (choice)
        {
```

```
case 1:
    printf("\n Enter the key to be inserted:");
    scanf("%d", &k);
    root = insert(root, k);
    break;

case 2:
    printf("\n");
    display(root, 0);
    printf("\n");
    break;

case 3:
    ptr = Min(root);
    if (ptr != NULL)
        printf("Min key is %d\n", ptr->info);
    ptr = Max(root);
    if (ptr != NULL)
        printf("Max key is %d\n", ptr->info);
    break;

case 4:
    exit(1);

default:
    printf("\n Wrong choice");
}
}
return 0;
}
```

```
struct node * insert (struct node *ptr, int ikey)
{
    if (ptr==NULL)
    {
        ptr= (struct node*) malloc( sizeof (struct node));
        ptr->info = ikey
        ptr->lchild = NULL;
        ptr->rchild = NULL;
        else if (ikey < ptr->info)
            ptr->lchild = insert (ptr->lchild, ikey);
        else if (ikey > ptr->info)
            ptr->rchild = insert (ptr->rchild, ikey);
        else
            printf ("\n Duplicate key\n");
            return ptr;
    }
struct node * Max (struct node* ptr)
{
    if (ptr==NULL)
        return NULL;
    else if (ptr->rchild ==NULL)
    return ptr;
    else
    return Max (ptr->rchild);
    }
```

```
struct node * Min (struct node * ptr)
{
   if (ptr==NULL)
     return NULL;
   else if (ptr -> lchild ==NULL)
     return ptr;
   else
     return Min (ptr -> lchild);
   }

void display (struct node * ptr, int level)
{
   int i;
   if (ptr == NULL)
       return;
   else
   {
     display (ptr -> rchild, level+1);
     printf ("\n");
     for (i=0; i< level; i++)
       printf ("  ");
     printf ("%d", ptr -> info);
     display (ptr -> lchild, level + 1);
   }
}
```