

## Data Structure Lab Test -2

**3. Write a program to construct a Binary Search Tree and also to find the maximum value in a Binary Search Tree.**

**Solution: -**

```
#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 *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 maximum\n");
    printf("4.Quit\n");
    printf("\nEnter your choice : ");
    scanf("%d",&choice);

    switch(choice)
    {
    case 1:
        printf("\nEnter 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 = Max(root);
```

```
if(ptr!=NULL)
```

```
printf("\nMaximum key is %d\n", ptr->info );
```

```
break;
```

```
case 4:
```

```
exit(1);
```

```
default:
```

```
printf("\nWrong choice\n");
```

```
}
```

```
}
```

```
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) /*Insertion in left subtree*/
        ptr->lchild = insert(ptr->lchild, ikey);
    else if(ikey > ptr->info) /*Insertion in right subtree */
        ptr->rchild = insert(ptr->rchild, ikey);
    else
        printf("\nDuplicate 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);
}

```

```

void display(struct node *ptr,int level)

```

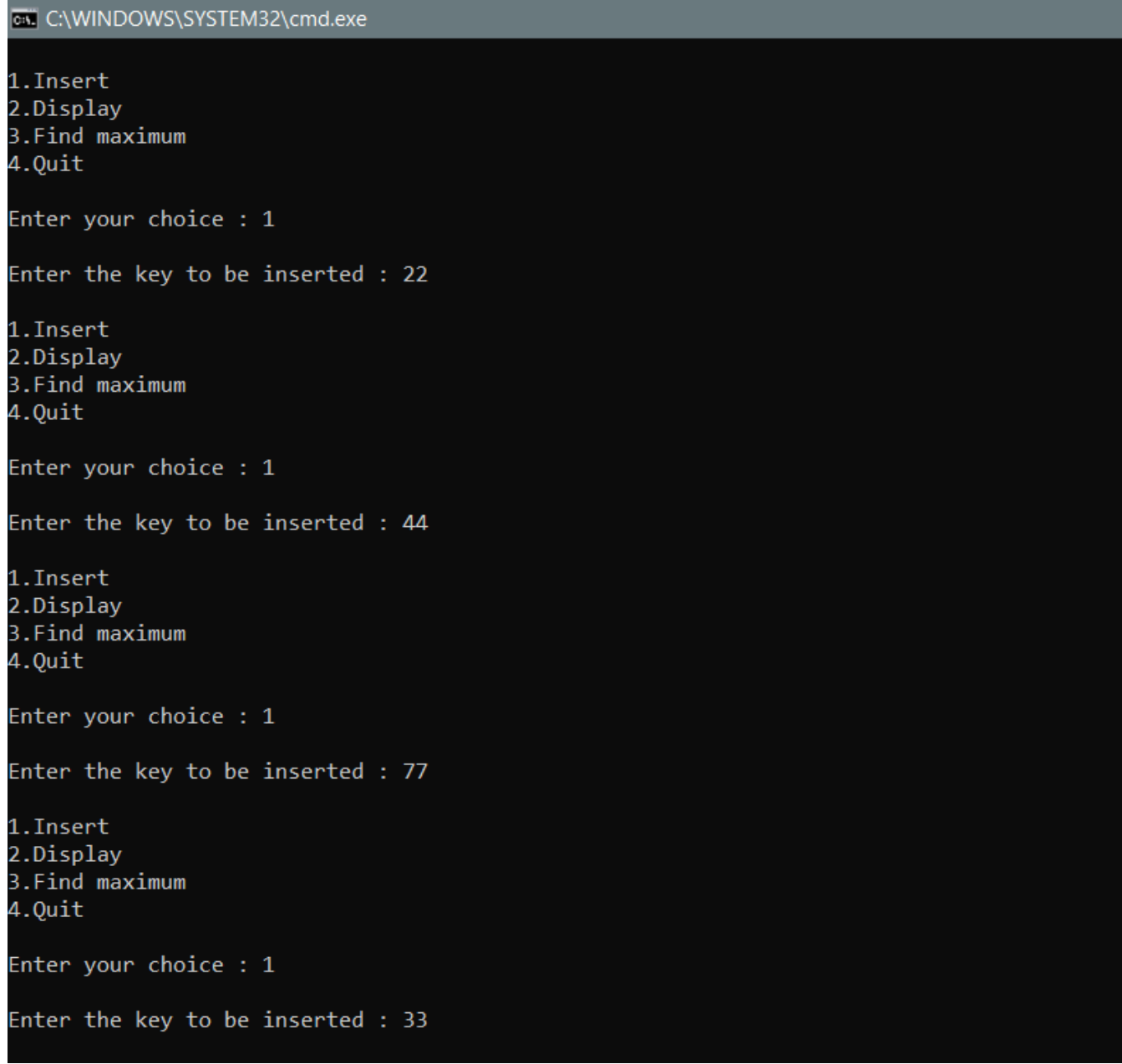
```

{
    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);
    }
}

```

```
}  
}
```

## Output: -



```
C:\WINDOWS\SYSTEM32\cmd.exe  
  
1.Insert  
2.Display  
3.Find maximum  
4.Quit  
  
Enter your choice : 1  
  
Enter the key to be inserted : 22  
  
1.Insert  
2.Display  
3.Find maximum  
4.Quit  
  
Enter your choice : 1  
  
Enter the key to be inserted : 44  
  
1.Insert  
2.Display  
3.Find maximum  
4.Quit  
  
Enter your choice : 1  
  
Enter the key to be inserted : 77  
  
1.Insert  
2.Display  
3.Find maximum  
4.Quit  
  
Enter your choice : 1  
  
Enter the key to be inserted : 33
```

Enter your choice : 2

```

    77
  44
    33
22
```

1.Insert  
2.Display  
3.Find maximum  
4.Quit

Enter your choice : 3

Maximum key is 77

1.Insert  
2.Display  
3.Find maximum  
4.Quit

Enter your choice : 4

-----

(program exited with code: 1)

Press any key to continue . . .