

Name:- Alok Kumar Rastogi

Page:- (1)

Sem:- 3

USN:- 1BM19CS192

Subject:- DS Lab Test-2

Date:- 4/01/2021

Alok

Q Write a program to Construct a Binary Search Tree and also to find the maximum value in a Binary search Tree.

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

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

```
struct node
```

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

```
struct node *insert (struct node *ptr, int key);
```

```
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 ("1.Insert\n 2.Display\n 3.Find maximum\n 4.Quit\n");
```

```
    printf ("Enter your choice: ");
```

```
    scanf ("%d", &choice);
```

Switch (choice)

{

Case 1:

printf("Enter two key to be inserted!");

scanf("%d %d", &K);

root = @insert(root, K);

break;

Case 2:

printf("In");

display(root, 0);

printf("In");

break;

Case 3:

ptr = Max(root);

if (ptr != NULL)

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

break;

Case 4:

exit(1);

default:

printf("Wrong 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;
}
```

elseif (ikey < ptr -> info)

ptr -> lchild = insert (ptr -> lchild, ikey);

elseif (ikey > ptr -> info)

ptr -> rchild = insert (ptr -> rchild, ikey);

else

printf ("In 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);

```
}
```

```
void display (struct node * ptr, int level)
```

```
{
```

```
    int i;
```

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

```
        return;
```

```
    else
```

```
    { display (ptr->rchild, level+1);
```

```
      printf ("ln");
```

```
      for (i=0; i<level; i++)
```

```
          printf (" ");
```

```
      printf ("%d ", ptr->info);
```

```
      display (ptr->lchild, level+1);
```

```
    }
```

```
}
```