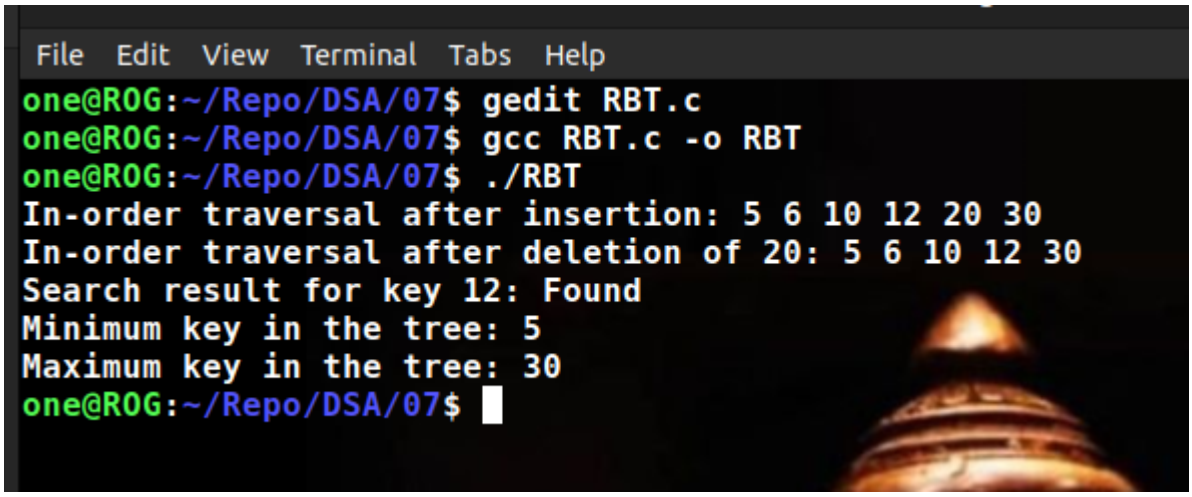
Data Structures and Algorithms

Assignment - VIII

INPUT:

```c
int main() {
    splay_tree *t = new_splay_tree();

    node *a, *b, *c, *d, *e, *f, *g, *h, *i, *j, *k, *l, *m;
    a = new_node(10);
    b = new_node(20);
    c = new_node(30);
    d = new_node(100);
    e = new_node(90);
    f = new_node(40);
    g = new_node(50);
    h = new_node(60);
    i = new_node(70);
    j = new_node(80);
    k = new_node(150);
    l = new_node(110);
    m = new_node(120);

    insert(t, a);
    insert(t, b);
    insert(t, c);
    insert(t, d);
    insert(t, e);
    insert(t, f);
    insert(t, g);
    insert(t, h);
    insert(t, i);
    insert(t, j);
    insert(t, k);
    insert(t, l);
    insert(t, m);

    delete(t, a);
    delete(t, m);

    node *result = search(t, t->root, 50);
    if (result != NULL) {
        printf("Node found: %d\n", result->data);
    } else {
        printf("Node not found\n");
    }

    inorder(t, t->root);

    return 0;
}
```

OUTPUT:


File Edit View Terminal Tabs Help
one@ROG:~/Repo/DSA/07$ gedit RBT.c
one@ROG:~/Repo/DSA/07$ gcc RBT.c -o RBT
one@ROG:~/Repo/DSA/07$ ./RBT
In-order traversal after insertion: 5 6 10 12 20 30
In-order traversal after deletion of 20: 5 6 10 12 30
Search result for key 12: Found
Minimum key in the tree: 5
Maximum key in the tree: 30
one@ROG:~/Repo/DSA/07$