

Data Structures and Algorithms

Assignment - IX

INPUT:

```
int main() {
    Node *heap = NULL;

    // Insertion
    heap = insert(heap, 3);
    heap = insert(heap, 5);
    heap = insert(heap, 7);
    heap = insert(heap, 2);
    heap = insert(heap, 8);

    printf("Leftist Heap after insertion: ");
    inorderTraversal(heap);
    printf("\n");

    // Deletion
    heap = deleteMin(heap);

    printf("Leftist Heap after deletion: ");
    inorderTraversal(heap);
    printf("\n");

    return 0;
}
```

OUTPUT:

```
one@ROG:~/Repo/DSA/09$ gedit LH.c
one@ROG:~/Repo/DSA/09$ gcc LH.c -o lh
one@ROG:~/Repo/DSA/09$ ./lh
Leftist Heap after insertion: 5 3 7 2 8
Leftist Heap after deletion: 5 3 8 7
one@ROG:~/Repo/DSA/09$
```

File Edit View Terminal Tabs Help

one@R0G:~/Repo/DSA/07\$ gedit RBT.c

one@R0G:~/Repo/DSA/07\$ gcc RBT.c -o RBT

one@R0G:~/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@R0G:~/Repo/DSA/07\$