BHUSHITH GAGAN
1BM18CS024
28/10/2020

ADC LAB-6

INSERTION 2-3 TREE :

```
void Tree :: insert (int k) {
    if (root==NULL) {
        root = new Treenode (true);
        root -> keys [0] = k;
        root -> n = 1;
    }
    else {
        if (root -> n==3) {
            TreeNode *s = new TreeNode (false);
            s -> child [0] = root;
            s -> splitChild (0, root);
            int i=0;
            if (s -> keys[0]<k) i++;
            s -> child [i] -> insertNonfull (k);
            root = s;
        }
        else
            root -> insertNonfull (k);
    }
}
```

TRAVERSAL :

```
void TreeNode :: Traversal () {
    cout << "\n";
    int i;
    for (i=0; i<n; i++) {
        if (leaf == false) child[i] -> traverse();
        cout << " " << keys [i];
    }
```

BHUSHITH GAGAN
1BM18CS024

```
if (leaf == false)
        child[i] -> traverse ();
        Cout << " \n";
}
```

DELETION:

```
void TreeNode :: remove (int k) {
        int idx = findkey (k);
        if (idx < n && key [idx] == k) {
                if (leaf) remove_fromleaf (idx);
                else removeFromNonLeaf (idx);
        }
        else {
            if (leaf) {
                Cout << "The key doesn't exist \n";
                return;
            }
    bool flag = ((idx == n) ? true : false);
      if (child [idx] -> n < 2) fill (idx);
      if (flag && idx > n) child [idx - 1] -> remove (k);
      else   child [idx] -> remove (k);
      }
      return;
}
```