Nikhil.A.S
1BM18CS061

9664
25484
8182

B-Tree Insertion:

```
insert(int k)
    if(root == NULL)
        root = new node(t);
        root -> key[0]=k
        root -> n = 1;
    else
        if(root->n == 2*t-1)
            S= new node(t);
            S -> C[0] = root;
            S -> splitchild (0, root);
            i=0
            if(S->key[0] < k)
                i++;
            S -> C[i] -> insert nonfull(k);
            root = S
        else
            root -> insertNonFull (k)
insertNonFull(int k)
    i = n-1
    if (leaf == true)
        while (i>=0 && keys[i] >k)
            keys[i+1] = key[i];
            i++;
        keys[i+1] =k;
        n =n+1
```

```
else
  while(i>=0 && key[i]>K)
    i--;
  if(C[i+1]->n == 2*t-1)
     splitchild(i+1, C[i+1]);
     if(keys[i+1] <K)
        i++
  C[i+1] -> insertNonFull(K);


splitchild(int i, node *y)
  Z=new Node(y->t)
  Z->n = t-1
  for(int j=0; j<t-1; j++)
     Z->Key[j] = y->keys[j+t];
  if(y->leaf == false)
     for(int j=0; j<t; j++)
        Z->C[j] = y->C[j+t];
  y->n = t-1;

  for(j=n; j>=i+1; j--)
     C[j+1] = C[j].
  C[i+1]=Z
  for(j=n-1; j>=i; j--)
     keys[j+1 = keys[j];
  key[i] = y -> keys[t-1]
  n=n+1
```

Field Coordinator
Anand N
and