```
void insert into Node(int k) {
  int i = n-1;
  if (leaf) {
    while (i >= 0 && keys[i] > k) {
       keys[i+1] = keys[i];
       i--;
    }
    keys[i+1] = k;
     n = n+1;
  }
  else {
    while (i >= 0 && keys[i] > k)
          i--;
    if (C[i+1] -> n == 2*t - 1)
          split(i+1, C[i+1])
    ~~et~~ if (keys[i+1] < k)
          i++;
  }
    c[i+1] -> insertinto Node (k);
  }
}


void split(int i, ~~TwoThreeNode~~ BTree * y) {
BTree ~~TwoThreeNode~~ * z = new ~~TwoThreeNode~~ BTree (y->leaf);
  z~~y~~->n = t-1;
   for (int j = 0 ; j < t-1 ; j++)
     ~~[illegible]~~ z->keys[i] = 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 (int j=n ;j >=i+1; ~~j++)~~ j--)
    c[j+1] = c[j];

c[i+1] = z;
for (int j = n-1; j >= i ; j --)
      Keys [j+1] = Keys [j];
Keys [i] = y -> keys [t-1];
 n = n+1;
}
```