LAB

B-tree Insertion

```
def BTreeInsertion (T, k):

    root = self.root
    if (len(root.keys) == (2*self.t) - 1:

        temp = BTreeNode ()
        self.root = temp
        temp.child.insert (0, root)
        self.split_child (temp, 0)
        self.insert_nonfull (temp, k)
    else
        self.insert_non_full (root, k)

def insert_non_full (self, x, k):
    i = len (x.keys) - 1
    if (x.leaf)
        x.keys [i+1] = x.keys [i]
        while (i >= 0) and k[0] < x.keys [i]
            x.key [i+1] = x.keys [i]
            i = i-1
        x.keys [i+1] = k
    else
        while i >= 0 and k[0] < x.keys[i][0]
            i = i-1
        i = i+1
        if (len (x.child.keys) == (2*self.t) - 1:
        self.split_child (x, i)
```

```
if (k[0]) > x.keys[:][0]:
    i += 1
self.insert_non_full(x.child[i], k)
```