```
void
class B_treenode{
    int *keys;
    int m;
    B_treenode **child;
    int n;
    bool leaf;

};
friend class B tree{
        B_treenode *root;
        int m;
        void insert(int k)
  };

void B_tree :: insert (int k)
  {   int i;
      if root=NULL
        { root = new B_treenode(m, true);
          root->keys[0]= k
          root->n=1;
        }
      else {
        if (root->n=2*m-1)
          {
B_treenode *t=new B_treenode (m, false);
      t->child[0]-root;
      t -> split (0, root);
      i=0;
      if(t->keys[0]<k)   i++;
```

```
    t->child[i]->insertnonfull(k);
    root=t;
  }
  else
    root->nonfullinsert(k);
  }
}
void nonfullinsert(int k)
{ int i=n-1;
    if(leaf=true)
    { while(keys[i]>k && i>=0)
      {
        keys[i+1]=keys[i];
        i--;
      }
    keys[i+1]=k;
    n=n+1;
  }
  else { while(keys[i]>k && i>=0)  i--;
    if(child[i+1]->n == 2(m)-1)
    {
      Split(i+1, child[i+1]);
      if(keys[i+1]<k)
        i++;
    }
    child[i+1]->nonfullinsert(k);
  }
}
```