Name: Rohit Kudache USN: 1BM18CS083
V sem B' section
program 6 — Red-black Tree Insertion

```
void RBTree :: insert (const int &data) {
        Node *pt = new Node (data);
        root = BSTInsert (root, pt);
        fixViolation (root, pt);
    }

Node* BSTInsert (Node* root, Node *pt) {
        if (root == NULL)
                return pt;
        if (pt → data < root → data) {
            root → left = BSTInsert (root→left, pt);
            root → left → parent = root;
        }

        else if (pt → data > root → data) {
            root → right = BSTInsert (root → right, pt);
            root → right → parent = root;
        }
    return root;
}
```

case A : Parent of pt is left child of grand-parent of pt

case 1 : The uncle of pt is also red only -
- Recoloring required.

case 2 : pt is right child of it's parent left -
- rotation required.

case 3 : pt is leftchild of it's parent right -
- rotation required.

Case B : parent of pt is right child of Grand-parent of pt

case 1 : The uncle of pt is also red only recoloring required.

case 2 : pt is left child of it's parent Right rotation Requrred.

case 3 : pt is right child of it's parent then left rotation Required.

02