void BinTree::remove (int v) {
    Node * p = root, * pre, * pre2;
    int child_pos = -1;
    while (p && p->element != v) {
        pre = p;
        if (p->element < v) {
            p = p->right;
            child_pos = 1;
        } else {
            p = p->left;
            child_pos = 0;
        }
    }
    if (!p) {
        cout << " The unit is not found" << endl;
        return;
    }
    if (!p->left && !p->right) { //no replacement
        if (child_pos == -1) {
            root = NULL;
            delete p;
        } else {
            if (child_pos == 1) {
                pre->right = NULL;
                delete p;
            } else {
                pre->left = NULL;
                delete p;
            }
        }
        return;
    } else if (!p->right) {
        pre = p->left;
        if (!pre->right) {
            p->left = pre->left;
            p->element = pre->element;
            delete pre;
            return;
        }
        while (pre->right) {
            pre2 = pre;
            pre = pre->right;
        }
    }
}

```c
p-> element = pre->element;  
pre2-> right = pre->left;  
delete pre;  
}
else{  
pre = p -> right;  
if(!pre->left){  
p-> right = pre->right;  
p-> element = pre->element;  
delete pre;  
return;  
}  
while (pre -> left){  
pre2 = pre;  
pre = pre -> left;  
}  
p-> element = pre->element;  
pre2-> left = pre -> right;  
delete pre;  
}  
```