Harshit Hiremath,
IBM18CS036
16/12/2000

APS Lab 11 - Programs to

Binomial Heap writeup

Utility functions -

```
function revertList (Node *h) {
        if (h->sibling) {
                revertlist (h->sibling);
                h->sibling->sibling = h;
        }
        else  root = h;
}

function  findNode (Node *h, int val) {
        if (!h) return NULL;
        if (h->val == val) return h;
        Node *res = findNode (h->child, val);
        if (res != NULL) return res;
        return findNode (h->sibling, val);
}

fundion extractMinHeap (Node *h) {
        if (!h) return NULL;
        Node *min_prev = NULL;
        Node *min = h;
        int min = h->val;
        Node* curr = h;
        while (curr->sibling != NULL) {
                if (curr->sibling->val < min) {
                        min = curr->sibling->val;
                        min.prev = curr;
                        min = curr->sibling;
                }
```

```
        curr = curr->sibling;
    }
    if (min_prev = NULL && min->sibling == NULL) h = NULL;
    else if (min_prev != NULL) h=min->sibling;
    else min_prev->sibling = min_sibling;
    if (min->child) {
        reverseList(min->child);
        min->child->sibling = NULL;
    }
    return unionBHeap(h, root);
}

function delete(Node *h, int val){
    if(!h) return NULL;
    decreaseKeyBHeap(h, val, INT_MIN);
    return extractMinHeap(h);
}
function decreaseKeyBHeap(Node* h, int oldv, int newv) {
    Node* node= findNode(H, oldv);
    if (!node) return;
    Node->val = newV;
    Node *parent = Node->parent;
    while(parent != NULL && node->val < parent->val){
        swap(node->val, parent->val);
        node = parent;
        parent = parent->parent;
    }
}
```