Snehita J
1BM18CS107

```
#include <bits/stdc++.h>
using namespace std;
struct Node
{
    int val, degree;
    Node *p, *c, *s;
};
Node *root = NULL;
int binLink(Node *h1, Node *h2)
{
    h1->p = h2;
    h1->s = h2->c;
    h2->c = h1;
    h2->d = h2->d+1;
}
Node * createNode(int n)
{
    Node *new = new Node;
    new->val = n;
    new->p = NULL; new->s = NULL; new->c = NULL;
    return new;
}
Node * mergeHeaps(Node *h1, Node *h2)
{
    if(h1==NULL) return h2;
    Node *res = NULL;
    if(h1->d <= h2->d) res=h1;
  else if(h1->d > h2->d) res=h2;
  while(h1!=NULL && h2!=NULL)
  {
      if(h1->d < h2->d)
          h1 = h1->s;
      else if(h1->d == h2->d)
      {
          Node *sib = h1->s;
          h1->s=h2;
          h1=sib;
      }
  }
  return res;
```

Snehita. J
1BM18CS109

```
Node *binHeapDelete(Node *h, int val)
{
    if (h == NULL) return NULL;
    decreaseKeyBHeap(h, val, INT_MIN);
    return extractMin(h);
}

void decreaseKeyBHeap(Node *H, int old_val, int new)
{
    Node *n = findNode(H, old_val);
    if (node == NULL) return;
    node->val = new_val;
    Node *p = node->p;
    while (p != NULL && node->val < p->val)
    {
        swap(node->val, p->val);
        node = p;
        p = p->p;
    }
}

Node * findNode(Node *h, int val)
{
    if (h == NULL) return NULL;
    if (h->val == val) return h;
    Node *res = findNode(h->c, val);
    if (res != NULL) return res;
    return findNode(h->s, val);
}
```

```
int main()
{
    binHeapInsert(10)
    binHeapInsert(20), 30, 40, 50;
    root = binHeapDelete(root, 10);
    display(root);
}
```

Suchila J
1BM18CS109