Prog-10 : Binomial Heap opers,
(i) Delete (H) (ii) decrease key(H)

```
// Decrease key
Void decreseKeyBHeap (Node *H, int old_value,
    int new_value)
{
  // Check element is present (or) not
   Node *node = findNode (H, old_val);
  // return if node is not present
   if (node == NULL)
     return;
  // reduce the value to min.
   node -> val = new_val;
   Node * parent = node -> parent;
  // update the heap acc to reduced val
  while (parent != NULL && node -> val < parent -> val)
{
  Swap (node -> val, parent -> val);
   node = parent.
   parent = parent -> parent;
}
}
// Delete

Node * binomialHeap Delete (Node *h, int val)
{
  // check if heap is empty / not
   if (h == NULL)
      return NULL;
  // Reduce the val of ele to min
   decreaseKeyBHeap (h, val, INT_MIN);
  // Delete min ele from heap
   return extractMinBHeap (h);

}
```