

#10

```
void decreaseKey BHeap (Node *H, int old, int new)
{ Node *node = FindNode (H, old);
  if (node == NULL)
```

```
    return;
```

```
    Node → val = new;
```

```
    Node *parent = node → parent;
```

```
    while (parent && parent → val > node → val)
```

```
    { swap (node → val, parent → val);
```

```
      node = parent;
```

```
      parent = parent → parent;
```

```
    }
```

```
}
```

```
Node * heapDelete (Node *H, int val) {
```

```
    if (!H)
```

```
        return null;
```

```
    decreaseKey BHeap (H, val, INT_MIN);
```

```
    return Extract N in BHeap (H);
```

```
struct Node {
```

```
    int val, degree;
```

```
    Node *parent, *child, *sibling;
```

```
    int key;
```

```
};
```

```
Rval = key;  
degree = 0;  
child = sibling = parent = NULL;  
}
```

```
}
```