Program 4

Insertion

```
struct node* insert (struct node *r, int data)

  { if (r==NULL)
    { struct node *n;
      n= new struct node;
      n -> data = data;
      r = n;
      r -> left = r -> right=NULL;
      r -> height =1;
      return r.
    }
else
{
   if (data < r->data)
      r ->left = insert (r->left, data);
   else
      r -> right = insert (r->right, data);

}
      r -> height = calheight (r);
   if (bf(r) ==2 && bf(r->left)==1){
         r= LLrotation (r)
   }
else if (bf(r)==-2 & & bf(r->right)==-1)
 { r= rrotate(r);
 }
  else if (bf(r)= = -2&& bf (r-> right)==1)
  r = rrotationr(r)
 }
else if (bf(r)==2 && bf(r->left)==-1)
         { r = lrrotate(r);
      }
      return r;
    }

Struct node* deleteNode (Struct node* p, int data)
   { if(p-> left==NULL&& p->right==NULL)
        { if (p == this->root)
             this root = NULL
             delete p;
           return NULL
        }
        Struct node *t;
        Struct node *q;
        if (p ->data < data)
          { p->right = deleteNode (p->right, data);
        }
```

```
else if
{
  if (p -> left != NULL;
  { q = inpre (p -> left);
     p -> data = q -> data;
    p -> left = delete Node (p -> left, q -> data);
  }
  else
  { if (p -> left != NULL)
    { q = inpre (p -> left);
     q -> data = p -> data;
     p -> left = delete Node (p -> left, q -> data);
    }
    else
    { q = insuc (p -> right);
        p -> data = q -> data
      p -> right = delete Node (q -> right, q -> data);
    }
  }
if (bf (p == 2 && bf (p -> left == 1) { p = llrotation (p);
else if ((bf(p) == 2 && bf (p -> left) == -1) { p =
                                           lrrot (p);
else if ((bf(p) == 2 && bf (p -> left) == 0)
    p = llrotation (p);

return p;
```