```
typedef struct Node *  ref;

struct Node {
   int  key;
   int  count;
   int  bal;
   ref  left;
   ref  right;
};
/* ---------------------------------------- */
void searchAdd(int x, ref &p, int &h) {
   ref  p1, p2;

   if (p == NULL) {
      h = 1;
      p = new Node;
      p->key   = x;
      p->count = 1;
      p->bal   = 0;
      p->left  = p->right = NULL;
   }
   else
      if (x < p->key) {
         searchAdd(x, p->left, h);

         if (h)
            switch (p->bal) {
               case 1:
                  p->bal  = 0;
                  h       = 0;
                  break;

               case 0:
                  p->bal = -1;
                  break;

               case -1:
                  p1 = p->left;

                  if (p1->bal == -1) {          // LL
                     p->left   = p1->right;
                     p1->right = p;

                     p->bal  = 0;
                     p       = p1;
                  }
                  else {                        // LR
                     p2 = p1->right;

                     p1->right = p2->left;
                     p2->left  = p1;
```

```
            p->left   = p2->right;
            p2->right = p;

            if (p2->bal == -1)  p->bal = 1;
            else                p->bal = 0;

            if (p2->bal == 1)   p1->bal = -1;
            else                p1->bal = 0;

            p = p2;
         }

         p->bal  = 0;
         h       = 0;
      }
   }
   else
      if (x > p->key) {
         searchAdd(x, p->right, h);

         if (h)
            switch (p->bal) {
               case -1:
                  p->bal  = 0;
                  h       = 0;
                  break;

               case 0:
                  p->bal = 1;
                  break;

               case 1:
                  p1 = p->right;

                  if (p1->bal == 1) {         // RR
                     p->right = p1->left;
                     p1->left = p;

                     p->bal  = 0;
                     p       = p1;
                  }
                  else {                      // RL
                     p2 = p1->left;

                     p1->left  = p2->right;
                     p2->right = p1;

                     p->right = p2->left;
                     p2->left = p;

                     if (p2->bal == 1)   p->bal = -1;
```

```
                    else                p->bal = 0;
                    if (p2->bal == -1)  p1->bal = 1;
                    else                p1->bal = 0;

                    p = p2;
                 }
                 p->bal  = 0;
                 h       = 0;
              }
           }
           else {
              p->count ++;
              h = 0;
           }
}
/* ---------------------------------------- */
void balance1(ref &p, int &h) {
   ref  p1, p2;
   int  b1, b2;

   switch (p->bal) {
      case -1:
         p->bal = 0;
         break;

      case 0:
         p->bal  = 1;
         h       = 0;
         break;

      case 1:
         p1 = p->right;
         b1 = p1->bal;

         if (b1 >= 0) {           // RR
            p->right = p1->left;
            p1->left = p;

            if (b1 == 0) {
               p->bal  = 1;
               p1->bal = -1;
               h       = 0;
            }
            else {
               p->bal  = 0;
               p1->bal = 0;
            }
            p = p1;
         }
         else {                   // RL
            p2 = p1->left;
```

```
            b2 = p2->bal;
            p1->left  = p2->right;
            p2->right = p1;

            p->right = p2->left;
            p2->left = p;

            if (b2 == 1)   p->bal = -1;
            else           p->bal = 0;

            if (b2 == -1)  p1->bal = 1;
            else           p1->bal = 0;

            p      = p2;
            p2->bal = 0;
         }
      }
   }
   /* ---------------------------------------- */
   void balance2(ref &p, int &h) {
      ref  p1, p2;
      int  b1, b2;

      switch (p->bal) {
         case 1:
            p->bal = 0;
            break;

         case 0:
            p->bal  = -1;
            h       = 0;
            break;

         case -1:
            p1 = p->left;
            b1 = p1->bal;

            if (b1 <= 0) {                // LL
               p->left   = p1->right;
               p1->right = p;

               if (b1 == 0) {
                  p->bal  = -1;
                  p1->bal = 1;
                  h = 0;
               }
               else {
                  p->bal  = 0;
                  p1->bal = 0;
               }
               p = p1;
```

```
        }
        else {                          // LR
          p2 = p1->right;
          b2 = p2->bal;

          p1->right = p2->left;
          p2->left  = p1;
          p->left   = p2->right;
          p2->right = p;

          if (b2 == -1)  p->bal = 1;
          else           p->bal = 0;

          if (b2 == 1)   p1->bal = -1;
          else           p1->bal = 0;

          p      = p2;
          p2->bal = 0;
        }
   }
}
/* ---------------------------------------- */
void del(ref &q, ref &r, int &h) {
   if (r->right) {
      del(q, r->right, h);

      if (h)
         balance2(r, h);
   }
   else {
      q->key    = r->key;
      q->count  = r->count;
      q         = r;
      r         = r->left;
      h = 1;
   }
}
/* ---------------------------------------- */
void searchDelete(int x, ref &p, int &h) {
   ref  q;
   if (p == NULL)   // Không có
      h = 0;
   else
      if (x < p->key) {
         searchDelete(x, p->left, h);
         if (h)
            balance1(p, h);
      }
      else
         if (x > p->key) {
            searchDelete(x, p->right, h);
```

```
            if (h)
               balance2(p, h);
         }
         else {
            q = p;
            if (q->right == NULL) {
               p = q->left;
               h = 1;
            }
            else
               if (q->left == NULL) {
                  p = q->right;
                  h = 1;
               }
               else {
                  del(q, p->left, h);
                  if (h)
                     balance1(p, h);
               }

            delete q;
         }
}
```