## 题目描述：

## 在一个排序的链表中，存在重复的结点，请删除该链表中重复的结点，重复的结点不保留，返回链表头指针。 例如，链表1->2->3->3->4->4->5 处理后为 1->2->5

## 解题思路

用三个指针。pre用于保存当前结点的前一结点，p用于保存当前结点，q用来表示与当前结点p val值相同的最后一个节点，然后pre->next = q->next，就将重复值删除掉。

特例： 1->1->1; 需要判断头结点是否和当前结点p相同 if（phead == q）

算法描述：

/\*

struct TreeLinkNode {

int val;

struct TreeLinkNode \*left;

struct TreeLinkNode \*right;

struct TreeLinkNode \*next;

TreeLinkNode(int x) :val(x), left(NULL), right(NULL), next(NULL) {

}

};

\*/

class Solution {

public:

TreeLinkNode\* GetNext(TreeLinkNode\* pNode)

{

if(pNode == NULL)

return pNode;

if(pNode->right != NULL)

{

TreeLinkNode \*p = pNode->right;

while(p->left != NULL)

{

p = p->left;

}

return p; // 右子树存在，则返回右子树的最左节点

}

while(pNode->next != NULL)

{

TreeLinkNode \*p = pNode->next;

if(p->left == pNode)

return p; //右子树不存在，存在父节点，且为父节点的左节点

pNode = pNode->next;//右子树不存在，存在父节点，为父节点的右节点

}

return NULL;

}

};