## 1 题目

反转一个单链表。

**示例:**

**输入:** 1->2->3->4->5->NULL

**输出:** 5->4->3->2->1->NULL

struct ListNode

{

int val;

ListNode \*next;

ListNode(int x) : val(x), next(NULL) {}

};

ListNode\* reverseList(ListNode\* head)

{

}

## 2 算法分析设计

1. 定义两个指针指向要操作的节点，P1和P2。



(2)Head->next指向空，P2->next指向前一个



1. P1和P2向后移动一步，继续执行步骤（2），直到P2为空，退出循环。



## 3 源代码分析

ListNode \* ListReverse::reverseList(ListNode \* head)

{

//链表位空返回空指针

if (head == nullptr)

{

return nullptr;

}

//链表只有一个节点，返回当前节点

if (head->next == nullptr)

{

return head;

}

ListNode \*p1 = head;

ListNode \*p2 = p1->next;

head->next = nullptr;

while (p2)

{

ListNode \*pTmp = p2->next;

p2->next = p1;

p1 = p2;

p2 = pTmp;

}

return p1;

}

## 4 测试用例

1. 输入空链表
2. 输入只有一个节点的链表
3. 输入多个节点的链表