

依次将后一个节点插入节点头，最后将head->next赋值null即可。



定义pre和next，pre->next=next，若节点val相等，next=next->next;pre->next=next;

不相等时直接同时向后遍历



定义cur和next，pre->next=next，若next->val与给定值相等，next=next->next;pre->next=next;

不相等时直接同时向后遍历



先得到两个链表中数据的个数，然后分别组合，得到数据后相乘



使用push\_back依次插入即可。



首先遍历得到链表节点数，保存为int类型，节点数/2+1即为中间节点。



比较l1.l2的val，按照大小依次赋值给链表即可



首先遍历得到链表节点数，取第（节点数-n）个节点 用时较长

修改：定义两个指针，先让一个移动n个节点，在同时移动，先移动的到达链表末尾时另一个指针指向倒数第n个、



定义两个指针，先让一个移动n个节点，若先移动的已经为null，则删除头节点，否则同时移动，先移动的到达链表最后一个节点另一个指针指向倒数第n+1个，让该节点指向下下个节点即可。



只给出了需要删除的节点，因此用该节点覆盖下一节点，删除下一个节点即可



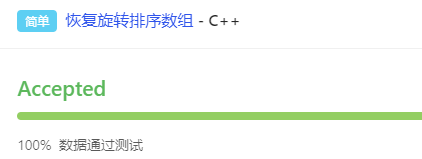
逐位相加，计算进位。循环相加使用carry计算进位，并将carry%10保存在链表中，令carry=carry/10,继续循环。



遍历head链表，逐个插入新链表。



使用二分法，左边界与右边界相加除以2得到mid，并根据mid位置的数值判断在左侧还是右侧。最后判断得到的数值是否与查找的相同，相同返回left或right对应数值，不同返回-1



首先找到最大数字与最小数字的交界处，取到后一个数字迭代器，然后运用reverse函数分别翻转前一组、后一组和整个容器即可。



两层循环遍历所有情况，得到结果后break即可、



首先判断vector是否为空，为空返回0，然后判断target是否小于vector中第一项，接着循环判断是否与数组中数相等或者在两项之间，相等返回i，在两项之间返回i+1，否则返回A.size()



先将数组合并，后排序。数组类似指针，直接在A后添加B的数值即可。



排序后将取下标(nums.size()-1)/2即可



定义ij，i循环增加，判断数组nums[i]否等于nums[j],不等时++j,并使nums[j]=nums[i],最后返回j即可。



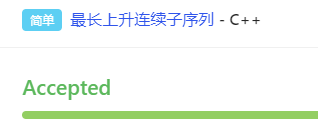
定义count来计算同一个数字重复的次数。



遍历，经过值为1的障碍时不计条数。



遍历，遇到值相同的就将后面所有数据前移。



因为正序逆序需要同时考虑，因此直接考虑正序上升和下降的序列，并进行比较。