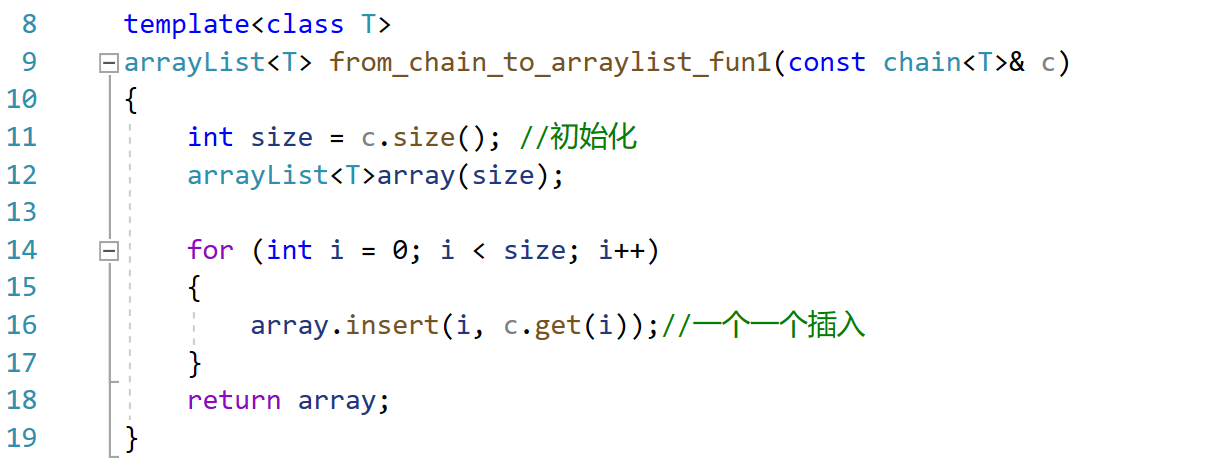
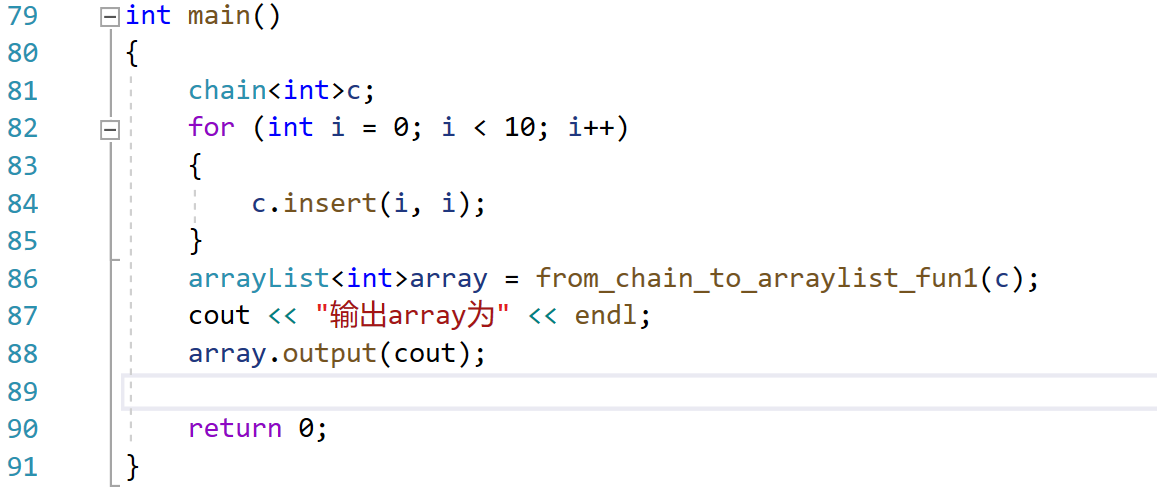
P124

**12题**

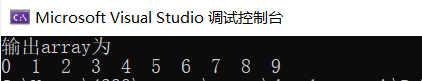
(1)



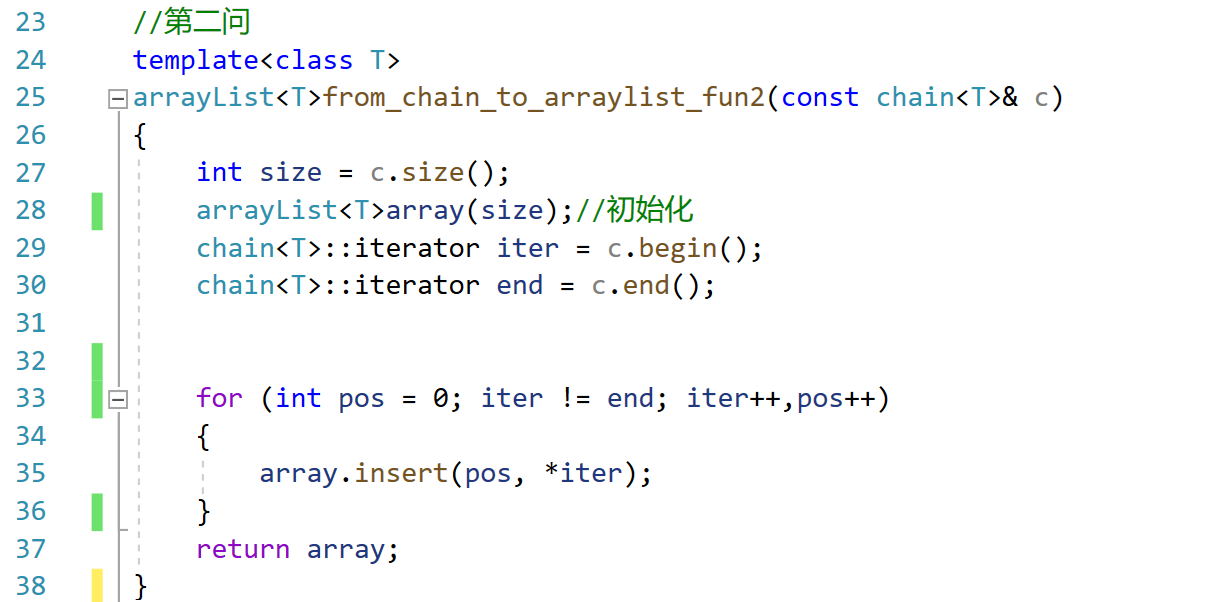
测试数据：



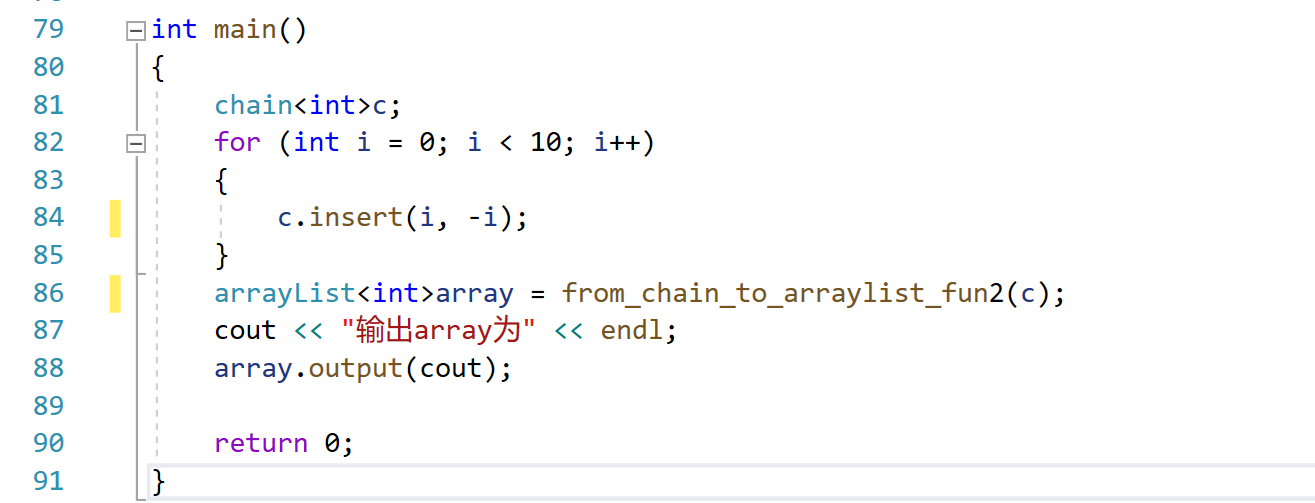
结果：（和预期结果相符）



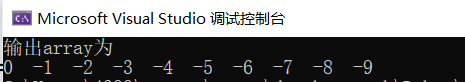
(2)



测试数据：



结果：（和预期结果相符）



**15题**

（1）



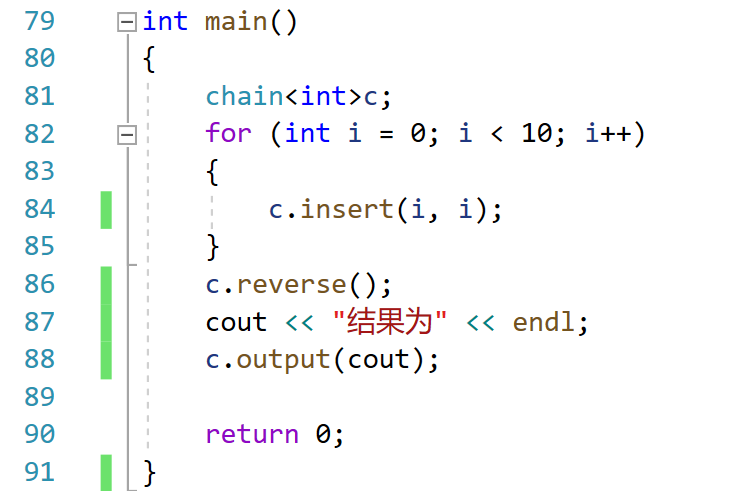
(2)

时间复杂度：因为我们使用了迭代器来进行操作，每一次的while循环，都会将p2的next指向p1所指向的chainNode，每次结束，迭代器都会指向下一个结点。当p2指向了nullptr，就结束，while循环内部是常数级，循环了n-1次，故时间复杂度为O(n)。

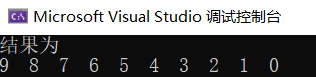
空间复杂度：因为我们没有分配新的结点，故空间复杂度O(1)(定义了迭代器).

(3)

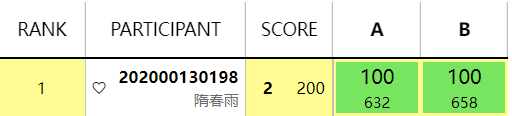
测试数据：



结果为：

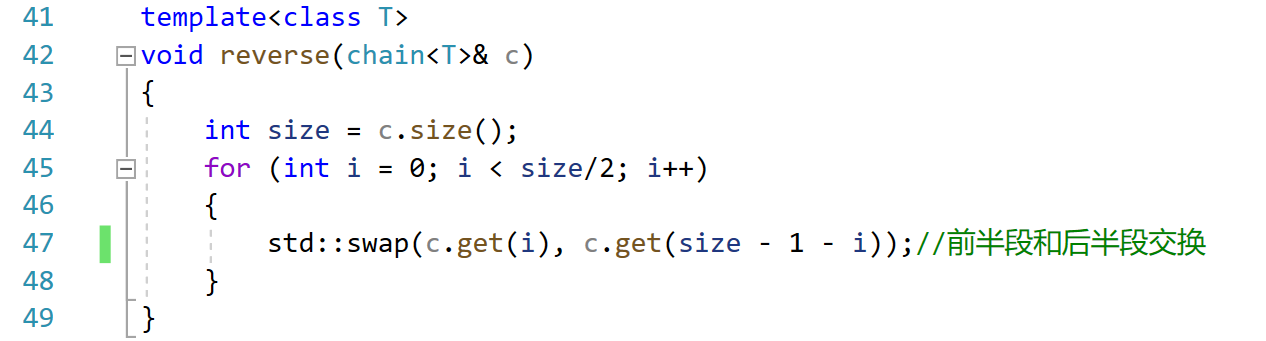


同时，实验中也有reverse的要求，提交后测试通过



**16**

**(1)**

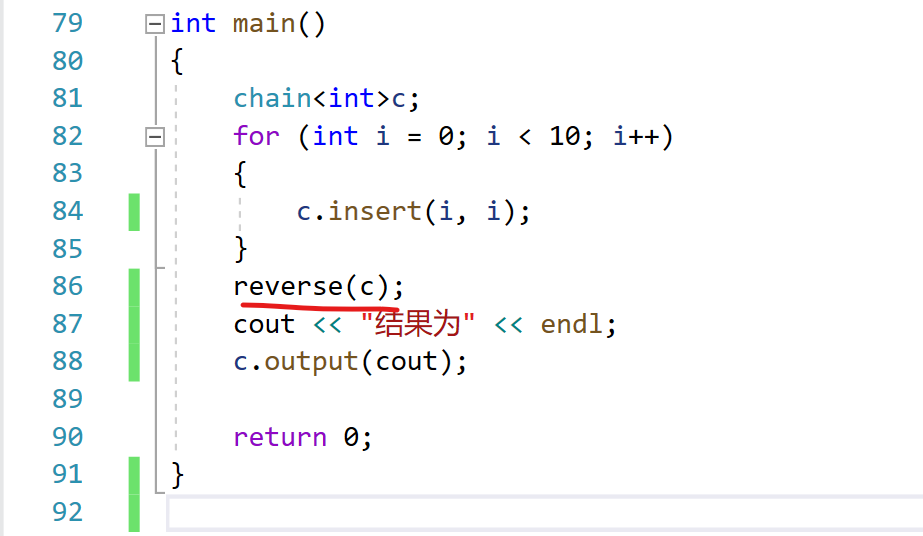
****

**(2)**

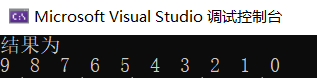
**时间复杂度：对于每一个链表结点的元素，都被get函数调用返回过，复杂度O（n）,std::swap进行了size/2次，单次复杂度O(1),总体时间复杂度O(n)**

**综上：时间复杂度O(n)**

**测试数据：**

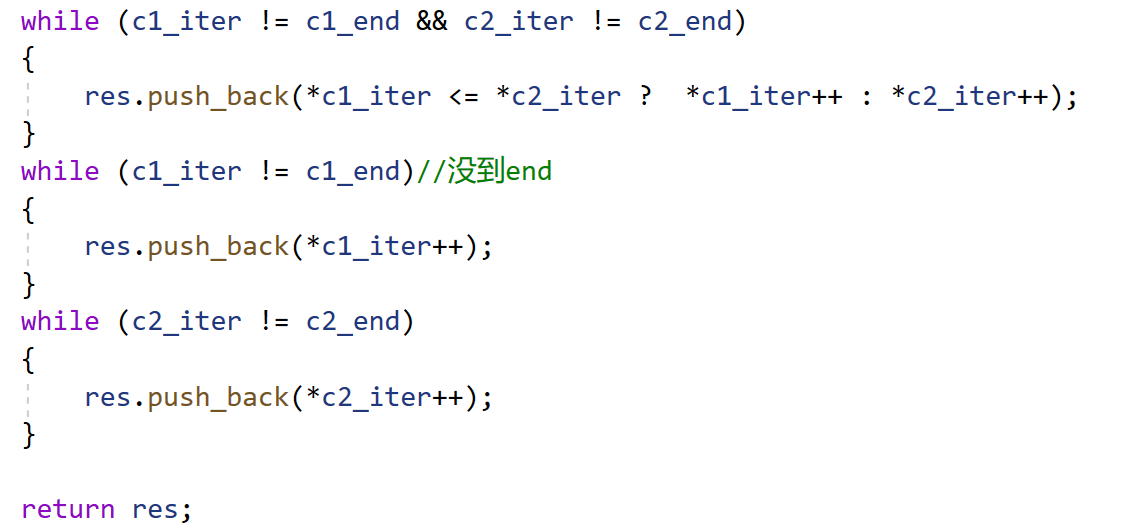
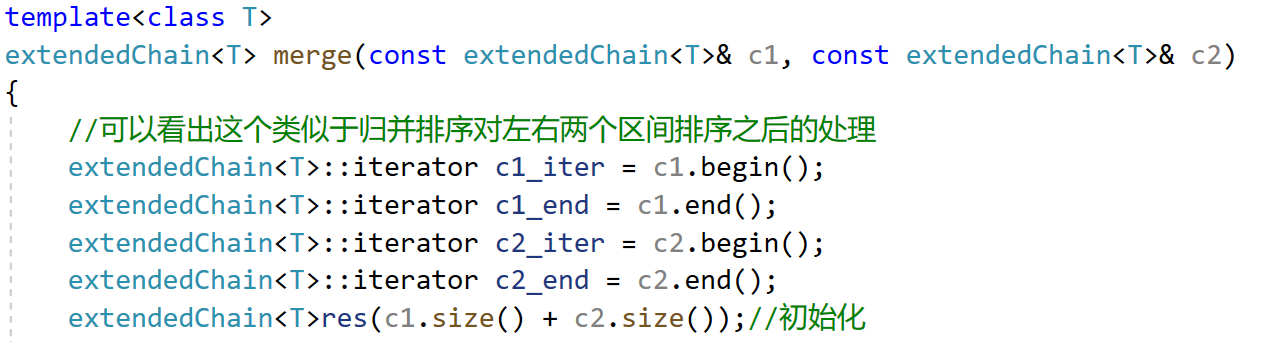
****

**结果：**

****

**19题**

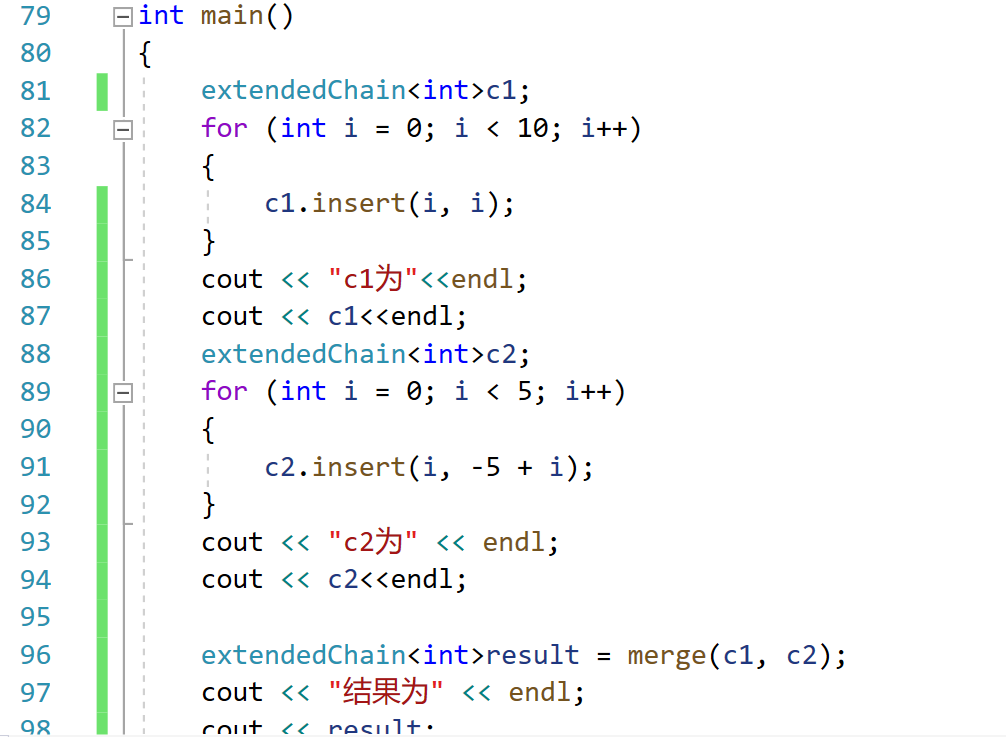
**(1)**

****

**(2)**

**时间复杂度：这个其实就是归并排序对左右两个区间排序之后的处理，时间复杂度O（n）**

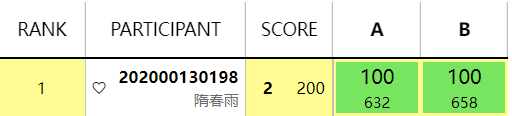
**(3)**

****

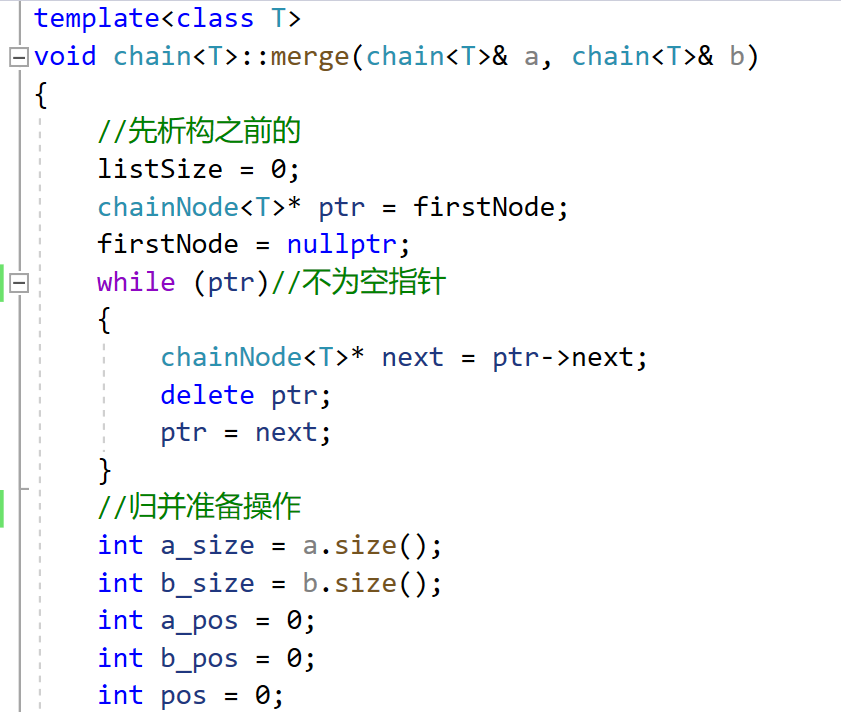
**结果：**

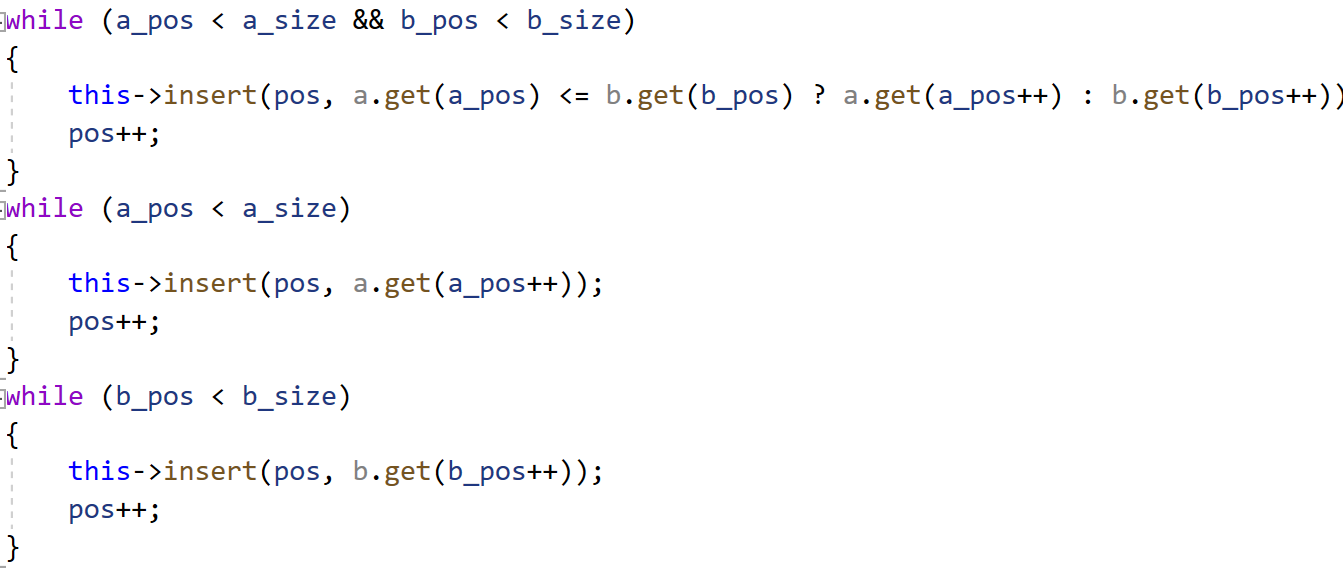
****

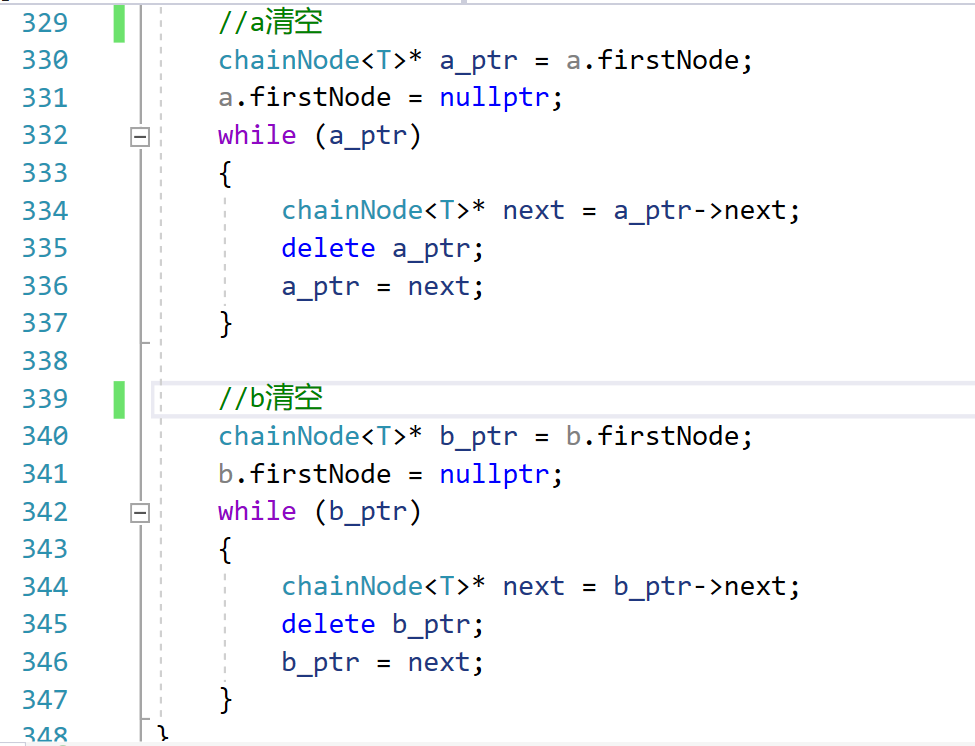
同时，实验中也有merge的要求，提交后测试通过



**20题**





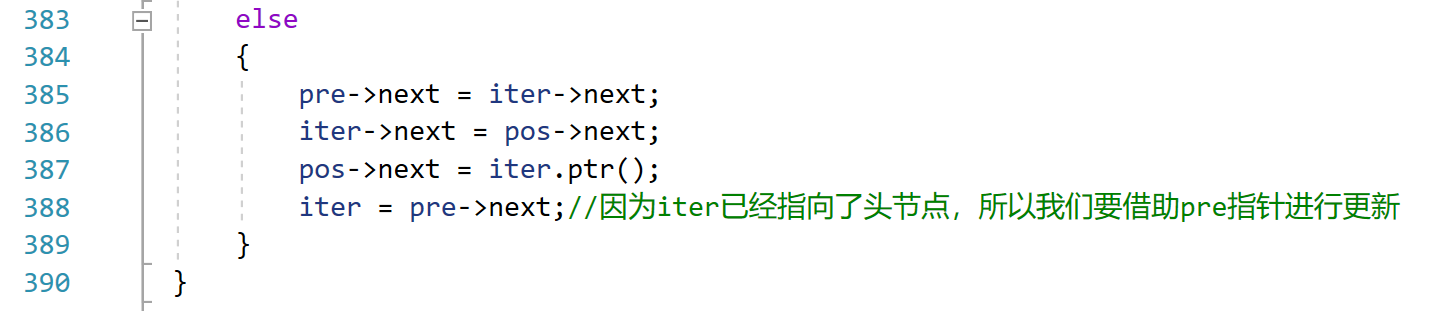
****

**26题**

**(1)**

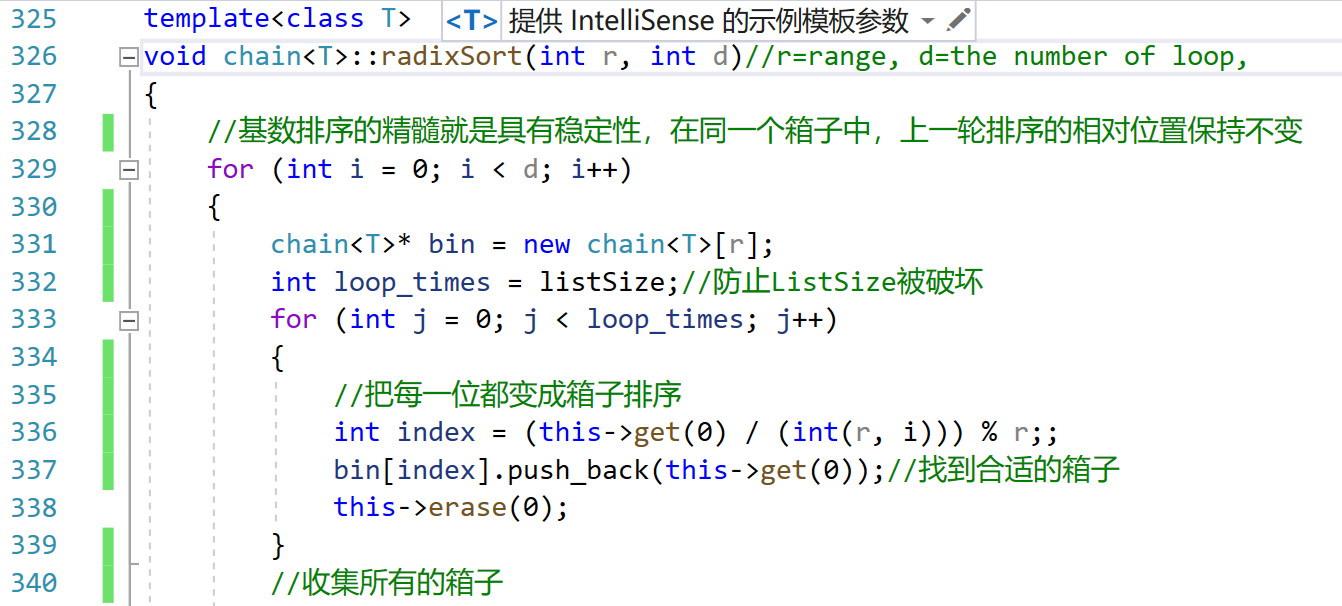
****

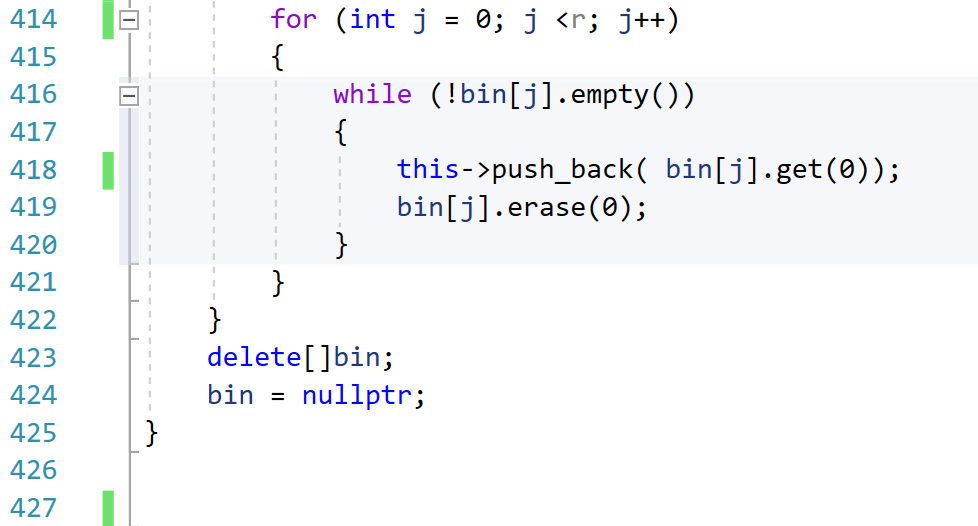
****

****

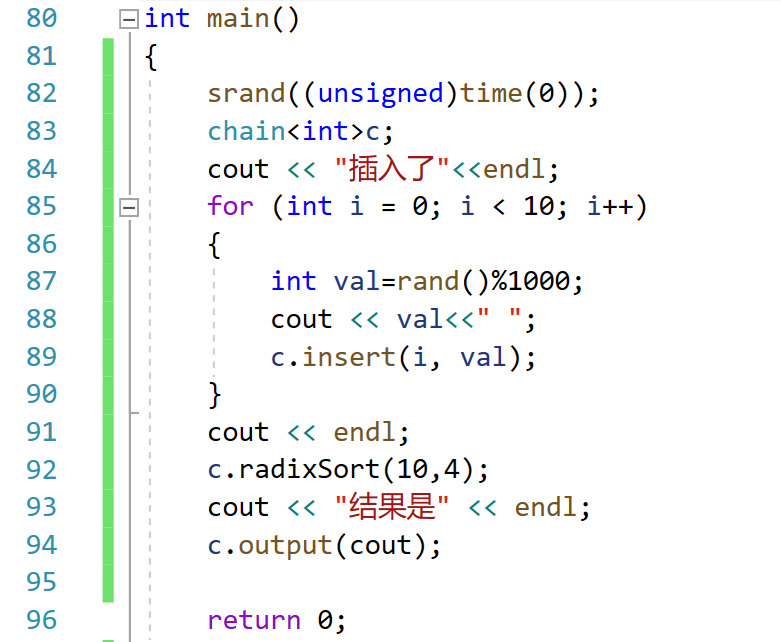
**64题**

1. 时间复杂度证明：d是基数排序分解的数字的个数，同时也是循环的次数。在每一次的循环中，我们需要建立r个箱子，并对其初始化，复杂度O（r）,还需要取出来链表的头结点的元素，然后将其放入对应的箱子中去，复杂度为O(n)。也就是说，基数排序建立在每一次的箱子排序之上，箱子排序的时间复杂度O(r+n)，循环d次，复杂度O(d(r+n))

****

****

**(2)测试数据**

****

**结果：**

****

**(3)**

**基数排序的时间复杂度为O（d(r+n)）,因为d是常数，所以为O（n），唯一的缺点就是空间复杂度较高，为O（n）。**

**我们使用for循环一百次来进行时间的计时（如果是单次，可能会有误差，并且clock可能过小测不出来）**

**经过测试**

**n=100时，插入排序时间为0.0027s,基数排序为0.0005s**

**n=1000时，插入排序时间为0.071s,基数排序为0.002s**

**n=10000时，插入排序时间为1.214s,基数排序为0.0724s**