1. 设计一个高效的算法，将顺序表的所有元素逆置，要求算法的空间复杂度为O（1）。
2. 从顺序表中删除具有最小值的元素并由函数返回被删除的元素值。空出的位置由最后一个元素填补，若顺序表为空则显示出错信息并退出运行。
3. 从顺序表中删除其值在给定值s和t之间(包含s和t)的所有元素，如果s或t不合理或者顺序表为空则显示出错信息并退出运行
4. 从有序顺序表中删除其值在给定s与t之间(要求s < t)的所有的元素，如果s或t不合理或者顺序为空则显示出错信息并退出运行。

注意本题和上一题的区别，因为是有序表，所有删除的元素必然是相连的整体。

1. 将两个有序表合并成一个新的有序顺序表，并由函数返回结果顺序表。
2. 从有序表中删除所有其值重复的元素，使表中所有元素的值均不同。
3. 长度为n的顺序表L，编写一个时间复杂度O(n)、空间复杂度为O(1)的算法，该算法删除线性表中所有值为x的数据元素。
4. 设将n(n > 1)个整数存放到一堆一维数组R中。试设计一个在时间和空间两个方面都尽可能高效的算法。将R中保存的序列循环左移p(0 < p < n)个位置，即将R中数据由(X0,X1,…Xn - 1)变换为(Xp, Xp+1,…,Xn-1, X0, X1,…,Xp - 1)。
5. 给出算法的基本思想
6. 根据思想,采用C或C++或Java语言描述算法，关键之处给出注释。
7. 说明你所设计的算法的时间复杂度和空间复杂度。
8. 一个长度为L(L >= 1)的升序序列S，处在第[L/2]个位置的数称为S的中位数。

例如，若序列S1 = {11, 13, 15, 17, 19},则S1的中位数是15，两个序列的中位数是包含它们所有元素的升序序列的中位数。

例如，若S2= {2, 4, 6, 8, 20}，则S1和S2的中位数是12。