**题目：假设一个顺序表L中的所有元素为整数，设计一个算法调整该顺序表，使其中所有小于零的元素放在所有大于零的元素前面**

函数：

1. **void** sortList(**int** a[], **int** n) {
2. **int**\* w = **new** **int**[n];
3. **int** i, k = 0;
4. **for** (i = 0; i < n; i++)
5. **if** (a[i] < 0) {
6. w[k] = a[i];
7. k++;
8. }
9. **for** (i = 0; i < n; i++)
10. **if** (a[i] > 0 || a[i] == 0) {
11. w[k] = a[i];
12. k++;
13. }
14. **for** (i = 0; i < k; i++) {
15. a[i] = w[i];
16. }
17. **delete**[]w;
18. }

时间复杂度：O(n)\_

空间复杂度：O(n)

验证：

1. #include<bits/stdc++.h>
2. **using** **namespace** std;
3. **void** sortList(**int** a[], **int** n);
5. **int** main() {
6. **int** a[8] = { 3,4,7,0,-6,2,-4,-1 };
7. sortList(a, 8);
8. **for** (**int** i = 0; i < 8; i++) {
9. cout << a[i] << " ";
10. }
11. }
12. **void** sortList(**int** a[], **int** n) {
13. **int**\* w = **new** **int**[n];
14. **int** i, k = 0;
15. **for** (i = 0; i < n; i++)
16. **if** (a[i] < 0) {
17. w[k] = a[i];
18. k++;
19. }
20. **for** (i = 0; i < n; i++)
21. **if** (a[i] > 0 || a[i] == 0) {
22. w[k] = a[i];
23. k++;
24. }
25. **for** (i = 0; i < k; i++) {
26. a[i] = w[i];
27. }
28. **delete**[]w;
29. }