

排序

同学们要自己动手，锻炼动手能力！

```
1 #include <stdio.h>
2
3 void insertion_sort(int *num, int n) {
4     for (int i = 1; i < n; i++) {
5         for (int j = i - 1; j >= 0; j--) {
6             if (num[j] > num[j + 1]) {
7                 int t = num[j];
8                 num[j] = num[j + 1];
9                 num[j + 1] = t;
10            } else {
11                break;
12            }
13        }
14    }
15 }
16
17 int main() {
18     int n, num[105];
19     scanf("%d", &n);
20     for (int i = 0; i < n; i++) {
21         scanf("%d", &num[i]);
22     }
23     insertion_sort(num, n);
24     for (int i = 0; i < n; i++) {
25         printf("%d ", num[i]);
26     }
27     printf("\n");
28     return 0;
29 }
```

```

1 #include <stdio.h>
2
3 void bubble_sort(int *num, int n) {
4     for (int i = 1; i < n; i++) {
5         int flag = 0;
6         for (int j = 0; j < n - i; j++) {
7             if (num[j] > num[j + 1]) {
8                 int t = num[j];
9                 num[j] = num[j + 1];
10                num[j + 1] = t;
11                flag = 1;
12            }
13        }
14        if (flag == 0) {
15            break;
16        }
17    }
18 }
19
20 int main() {
21     int n, num[105];
22     scanf("%d", &n);
23     for (int i = 0; i < n; i++) {
24         scanf("%d", &num[i]);
25     }
26     bubble_sort(num, n);
27     for (int i = 0; i < n; i++) {
28         printf("%d ", num[i]);
29     }
30     printf("\n");
31     return 0;
32 }

```

```

1 #include <stdio.h>
2
3 void quick_sort(int *num, int left, int right) {

```

```
4     if (left >= right) {
5         return ;
6     }
7     int t = num[left];
8     int l = left, r = right;
9     while (l < r) {
10         while (l < r && num[r] >= t) {
11             r--;
12         }
13         if (l < r) {
14             num[l] = num[r];
15         }
16         while (l < r && num[l] <= t) {
17             l++;
18         }
19         if (l < r) {
20             num[r] = num[l];
21         }
22     }
23     num[l] = t;
24     quick_sort(num, left, l - 1);
25     quick_sort(num, l + 1, right);
26 }
27
28 int main() {
29     int n, num[105];
30     scanf("%d", &n);
31     for (int i = 0; i < n; i++) {
32         scanf("%d", &num[i]);
33     }
34     quick_sort(num, 0, n - 1);
35     for (int i = 0; i < n; i++) {
36         printf("%d ", num[i]);
37     }
38     printf("\n");
39     return 0;
40 }
```

```
1 #include <stdio.h>
2
3 void selection_sort(int *num, int n) {
4     for (int i = 0; i < n - 1; i++) {
5         int ind = i;
6         for (int j = i + 1; j < n; j++) {
7             if (num[j] < num[ind]) {
8                 ind = j;
9             }
10        }
11        int t = num[i];
12        num[i] = num[ind];
13        num[ind] = t;
14    }
15 }
16
17 int main() {
18     int n, num[105];
19     scanf("%d", &n);
20     for (int i = 0; i < n; i++) {
21         scanf("%d", &num[i]);
22     }
23     selection_sort(num, n);
24     for (int i = 0; i < n; i++) {
25         printf("%d ", num[i]);
26     }
27     printf("\n");
28     return 0;
29 }
```

```
1 #include <stdio.h>
2
3 void merge_sort(int *num, int l, int r, int *temp) {
4     if (l >= r) {
5         return ;
6     }
7     int mid = (l + r) / 2;
8     merge_sort(num, l, mid, temp);
9     merge_sort(num, mid + 1, r, temp);
10    int p1 = l, p2 = mid + 1, n = r - l + 1;
11    for (int i = 0; i < n; i++) {
12        if (p2 > r || (p1 <= mid && num[p1] <= num[p2])) {
13            temp[i] = num[p1];
14            p1++;
15        } else {
16            temp[i] = num[p2];
17            p2++;
18        }
19    }
20    for (int i = 0, j = l; i < n || j <= r; i++, j++) {
21        num[j] = temp[i];
22    }
23 }
24
25 int main() {
26     int n, num[105];
27     scanf("%d", &n);
28     for (int i = 0; i < n; i++) {
29         scanf("%d", &num[i]);
30     }
31     int temp[105];
32     merge_sort(num, 0, n - 1, temp);
33     for (int i = 0; i < n; i++) {
34         printf("%d ", num[i]);
35     }
36     printf("\n");
37     return 0;
38 }
```