排序

同学们要自己动手，锻炼动手能力！

```c
#include <stdio.h>

void insertion_sort(int *num, int n) {
    for (int i = 1; i < n; i++) {
        for (int j = i - 1; j >= 0; j--) {
            if (num[j] > num[j + 1]) {
                int t = num[j];
                num[j] = num[j + 1];
                num[j + 1] = t;
            } else {
                break;
            }
        }
    }
}

int main() {
    int n, num[105];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &num[i]);
    }
    insertion_sort(num, n);
    for (int i = 0; i < n; i++) {
        printf("%d ", num[i]);
    }
    printf("\n");
    return 0;
}
```

```c
#include <stdio.h>

void bubble_sort(int *num, int n) {
    for (int i = 1; i < n; i++) {
        int flag = 0;
        for (int j = 0; j < n - i; j++) {
            if (num[j] > num[j + 1]) {
                int t = num[j];
                num[j] = num[j + 1];
                num[j + 1] = t;
                flag = 1;
            }
        }
        if (flag == 0) {
            break;
        }
    }
}

int main() {
    int n, num[105];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &num[i]);
    }
    bubble_sort(num, n);
    for (int i = 0; i < n; i++) {
        printf("%d ", num[i]);
    }
    printf("\n");
    return 0;
}
```

```c
#include <stdio.h>

void quick_sort(int *num, int left, int right) {
```

```c
    if (left >= right) {
        return ;
    }
    int t = num[left];
    int l = left, r = right;
    while (l < r) {
        while (l < r && num[r] >= t) {
            r--;
        }
        if (l < r) {
            num[l] = num[r];
        }
        while (l < r && num[l] <= t) {
            l++;
        }
        if (l < r) {
            num[r] = num[l];
        }
    }
    num[l] = t;
    quick_sort(num, left, l - 1);
    quick_sort(num, l + 1, right);
}

int main() {
    int n, num[105];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &num[i]);
    }
    quick_sort(num, 0, n - 1);
    for (int i = 0; i < n; i++) {
        printf("%d ", num[i]);
    }
    printf("\n");
    return 0;
}
```

```c
#include <stdio.h>

void selection_sort(int *num, int n) {
    for (int i = 0; i < n - 1; i++) {
        int ind = i;
        for (int j = i + 1; j < n; j++) {
            if (num[j] < num[ind]) {
                ind = j;
            }
        }
        int t = num[i];
        num[i] = num[ind];
        num[ind] = t;
    }
}

int main() {
    int n, num[105];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &num[i]);
    }
    selection_sort(num, n);
    for (int i = 0; i < n; i++) {
        printf("%d ", num[i]);
    }
    printf("\n");
    return 0;
}
```

```c
#include <stdio.h>

void merge_sort(int *num, int l, int r, int *temp) {
    if (l >= r) {
        return ;
    }
    int mid = (l + r) / 2;
    merge_sort(num, l, mid, temp);
    merge_sort(num, mid + 1, r, temp);
    int p1 = l, p2 = mid + 1, n = r - l + 1;
    for (int i = 0; i < n; i++) {
        if (p2 > r || (p1 <= mid && num[p1] <= num[p2])) {
            temp[i] = num[p1];
            p1++;
        } else {
            temp[i] = num[p2];
            p2++;
        }
    }
    for (int i = 0, j = l; i < n || j <= r; i++, j++) {
        num[j] = temp[i];
    }
}

int main() {
    int n, num[105];
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &num[i]);
    }
    int temp[105];
    merge_sort(num, 0, n - 1, temp);
    for (int i = 0; i < n; i++) {
        printf("%d ", num[i]);
    }
    printf("\n");
    return 0;
}
```