

## 20171011课后作业 参考答案

### A

选择排序的一个子任务。假定最小值与最大值都是第一个元素，后面逐个比较；若遇到更大/更小的元素则更新答案。输出最大数第一次出现的位置，只要判断当前值是不是**大于**max\_val值即可；输出最小数最后一次出现的位置，只要判断当前值是不是**小于等于**min\_val即可。

多组输入，以 0 表示结束。

```
#include <stdio.h>

int arr[1001];

int main() {
    int n;
    while (scanf("%d", &n) == 1 && n) {
        int i;
        for (i = 0; i < n; ++i) {
            scanf("%d", &arr[i]);
        }
        int minn_index = 0, maxx_index = 0;
        for (i = 1; i < n; ++i) {
            if (arr[i] > arr[maxx_index]) maxx_index = i;
            if (arr[i] <= arr[minn_index]) minn_index = i;
        }
        printf("%d %d\n", minn_index, maxx_index); // 记得输出这个换行
    }
    return 0;
}
```

### B

输入一个数组，问最少需要多少次交换**相邻**元素，可以使得整个数组按照升序排列。我们只需要统计一下冒泡排序过程中需要交换多少次就可以了。另外，如果存在下标*i, j*且对应的元素 $arr[i] > arr[j]$ ，那么肯定有一次交换是交换 $arr[i]$ 和 $arr[j]$ 的。于是，我们只需要统计有多少对这样的元素存在就行了。

```
#include <stdio.h>

int arr[1001];
int main() {
    int n;
    while (scanf("%d", &n) == 1) {
        int i, j;
        for (i = 0; i < n; ++i) {
            scanf("%d", &arr[i]);
        }
        int ans = 0;
        for (i = 0; i < n; ++i) {
            for (j = i + 1; j < n; ++j) {
                if (arr[i] > arr[j]) ans++;
            }
        }
        printf("%d\n", ans);
    }
    return 0;
}
```

### C

最少需要多少趟冒泡排序可以使得数组升序。比如对于数组1 3 4 2，经过一趟冒泡排序，交换4和2得到1 3 2 4；第二趟冒泡排序交换3和2，得到1 2 3 4；数组有序，因此只需要两趟冒泡排序即可使得数组1 3 4 2有序。

我们只需要判断一下在一趟中是否交换了元素即可；如果没有交换元素，那么代表数组已经是有序的。

```
#include <stdio.h>

int arr[1001];
int main() {
    int n;
    while (scanf("%d", &n) == 1) {
        int i, j;
        for (i = 0; i < n; ++i) scanf("%d", &arr[i]);
        int ans = 0; // 统计需要几趟排序
        for (i = 0; i < n; ++i) {
            int flag = 0; // 用于标记是否有元素发生交换
            ans++;
            for (j = 0; j < n - i - 1; ++j) {
                if (arr[j] > arr[j + 1]) {
                    flag = 1;
                    int tmp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = tmp;
                }
            }
            if (!flag) break; // 如果没有元素交换, 直接跳出循环即可
        }
        printf("%d\n", ans - 1);
    }
    return 0;
}
```

## D

需要注意的是插入之后数组保持**有序**, 那么可以是升序也可以是降序。最靠前和最靠后的位置不同, 是因为数组中有大于等于一个与待插入数字相同的数字存在。

```
#include <stdio.h>

int arr[1001];
int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int n, i;
        scanf("%d", &n);
        for (i = 0; i < n; ++i) scanf("%d", &arr[i]);
        int tmp = arr[n - 1];
        int first = 0, second = 0;
        if (arr[0] < arr[n - 2]) { // 升序
            for (i = 0; i < n - 1; ++i) {
                if (arr[i] < tmp) first = i + 1; // 最后一个小于tmp值的下一个位置, 就是可以放的最靠前的
                if (arr[i] <= tmp) second = i + 1; // 最后一个小于等于tmp值的下一个位置, 就是可以放的最靠后的
            }
        } else {
            for (i = 0; i < n - 1; ++i) {
                if (arr[i] > tmp) first = i + 1;
                if (arr[i] >= tmp) second = i + 1;
            }
        }
        printf("%d %d\n", first, second);
    }
    return 0;
}
```

## E

按照绝对值排序, 任何一种排序方法都可以。注意输入整数的范围, 两个数相乘可能**超出int32**的表示范围。

```
#include <stdio.h>
#include <math.h> /* header file for function abs */

// 冒泡排序函数
void bubble_sort(int arr[], int n) {
    int i, j, tmp;
    for (i = 0; i < n; ++i) {
```

```
    for (j = 0; j < n - i - 1; ++j) {
        if (abs(arr[j]) < abs(arr[j + 1])) { // 根据绝对值来看是否要交换
            tmp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = tmp;
        }
    }
}

int main() {
    int n;
    int arr[111];
    while (scanf("%d", &n) == 1 && n) {
        int i;
        for (i = 0; i < n; ++i) {
            scanf("%d", &arr[i]);
        }
        bubble_sort(arr, n);
        for (i = 0; i < n - 1; ++i) {
            printf("%d ", arr[i]);
        }
        printf("%d\n", arr[n-1]);
    }
    return 0;
}
```