

# C++方向编程题答案

## 第一周

### day2

题目ID: 100448 排序子序列

链接: <https://www.nowcoder.com/questionTerminal/2d3f6ddd82da445d804c95db22dcc471?orderByHotValue=1&page=1&onlyReference=false>

#### 【题目解析】:

本题要求解的是排序子序列, 排序子序列为非递增或者非递减, 很多同学在这个非递增、非递减问题上很纠结, 注意: 非递减就是 $a[i] \leq a[i+1]$ , 递减就是 $a[i] > a[i+1]$ , 非递增就是 $a[i] \geq a[i+1]$ , 递增就是 $a[i] < a[i+1]$ 。其实这个不理解网上搜一下就理解了。

#### 【解题思路】:

解题思路:

1. 本题依次比较整个数组
2.  $a[i+1] > a[i]$ , 则进入非递增序列判断, 直到遍历到下一个值不大于等于为止count++, 然后进行下一位置的判断
3.  $a[i+1] < a[i]$ , 则进入非递增序列判断, 直到遍历到下一个值不小于等于为止count++, 然后进行下一位置的判断
4.  $a[i+1] == a[i]$ 不进行操作, ++i进行下一位置遍历, 因为相等既可以属于非递增序列, 也可以属于非递减序列。

本题注意点: 本题开始比较 $a[i+1]$ 与 $a[i]$ 进行比较, 为了避免越界, 数组定义为 $n+1$ 个, 同时给 $a[n] = 0$ ;

$a[n] = 0$ 带来的影响, 我们分为三种情况讨论:

1. 若到 $a[n-1]$ 的最后一组是非递减序列, 当 $i == n-1$ ,  $a[i] > a[i+1]$ , 因为前面的数都是大于0的, 这个输入条件已经说明了(去看看题目输入条件描述), 里面的循环结束,  $i++$ , count++,  $i == n$ , 外面的循环结束。
2. 若到 $a[n-1]$ 的最后一组是非递增序列, 当 $i == n-1$ ,  $a[i] > a[i+1]$ , 因为前面的数都是大于0的, 这个输入条件已经说明了(去看看题目输入条件描述), 循环再走一次,  $i++$ ,  $i == n$ , 里面的循环结束,  $i++$ , count++,  $i == n+1$ , 外面的循环结束。
3. 第三种情况 1 2 1 2 1 最后一个数是单独的情况, 后面补个0, 序列变成1 2 1 2 1 0, 当走完全面的序列  $i == n-1$ 时,  $a[i] > a[i+1]$ , 进入判断出一个非递增序列, count++,  $i++$ , 循环结束。
4. 也就是说数组最后一个位置多增加一个0, 不会影响第1、2情况的判断, 主要是帮助第3情况的正确判断。

```
#include<iostream>
#include<vector>
using namespace std;
```

// 本题牛客测试用例不全, 至少应该增加以下两组测试用例

// 输入:

// 4

```

// 1 3 2 3
// 输出: 2

// 输入:
// 6
// 3 2 1 1 2 3
// 输出: 2

int main()
{
    int n;
    cin >> n;

    // 注意这里多给了一个值, 是处理越界的情况的比较, 具体参考上面的解题思路
    vector<int> a;
    a.resize(n + 1);
    a[n] = 0;

    // 读入数组
    int i = 0;
    for (i = 0; i < n; ++i)
        cin >> a[i];

    i = 0;
    int count = 0;
    while (i < n)
    {
        // 非递减子序列
        if (a[i] < a[i + 1])
        {
            while (i < n && a[i] <= a[i + 1])
                i++;

            count++;
            i++;
        }
        else if (a[i] == a[i + 1])
        {
            i++;
        }
        else // 非递增子序列
        {
            while (i < n && a[i] >= a[i + 1])
                i++;

            count++;
            i++;
        }
    }

    cout << count << endl;

    return 0;
}

```

```
}
```

### 69389-倒置字符串

<https://www.nowcoder.com/practice/ee5de2e7c45a46a090c1ced2fdc62355?tpId=85&&tqId=29867&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking>

#### 【题目解析】：

本题题意很简单，就是将一段字符串中的前后单词交换，以单词为单位逆置。

#### 【解题思路1】：

先将整个字符串逆置过来，再遍历字符串，找出每个单词，对单词逆置。这里我们使用了stl算法中的reverse，所以这里使用迭代器遍历string

```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;

int main()
{
    string s;
    // 注意这里要使用getline, cin>>s遇到空格就接收结束了
    getline(cin, s);

    // 翻转整个句子
    reverse(s.begin(), s.end());
    // 翻转单词
    auto start = s.begin();
    while (start != s.end())
    {
        auto end = start;
        while (end != s.end() && *end != ' ')
            end++;

        reverse(start, end);

        if (end != s.end())
            start = end + 1;
        else
            start = end;
    }

    cout << s << endl;

    return 0;
}
```

#### 【解题思路2】：

第二思路是一个比较讨巧的思路，直接利用cin>>s接收输入，遇到空格就结束了，自然就分割开了每个单词，其次将每次接收到的单词拼接到之前串的前面就逆置过来了

```
#include <iostream>
#include <string>
using namespace std;
// cin读取string时自动会被空格分隔开，用另一个字符串存储进行逆序输出
int main()
{
    string s1, s2;
    cin >> s2;
    while (cin >> s1)
        s2 = s1 + " " + s2;
    cout << s2 << endl;
    return 0;
}
```