

# 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】:

通过上面理解了排序子序列, 如果我们把这里值画到一个坐标图里面, 本质就是去找这里图里面的波峰和波谷的个数, 再加一。

比如:

// 情况一: 在这里我们可以看到 1 2 3是一个递增序列, 也可以认为是一个非递减序列

// 在这里我们可以看到 2 2 1是一个非递增序列

```
      3
     2   2   2
    1       1
```

// 情况二: 需要注意这里如果去算波峰和波谷有两个, 但是要注意, 一个排序子序列至少需要有两个数构成,

// 所以这里的的排序子序列不是3个, 是两个: 1 3 和 2 3

```
      3       3
     2
    1
```

```
#include<iostream>
#include<vector>

using namespace std;
int main()
{
    int n;
    cin>>n;
    vector<int> v;
    v.resize(n);
    for(int i = 0; i < n; ++i)
        cin>>v[i];
```

```

int ret = 1;
for(int i=1; i<n-1; i++)
{
    //找出波峰波谷
    if((v[i-1] < v[i] && v[i] > v[i+1])
        || (v[i-1] > v[i] && v[i] < v[i+1]))
    {
        ret++;

        // 这里对应处理的就是上面情况二描述的问题
        if(i != n-3)
            i++;
    }
}
cout<<ret<<endl;
}

```

### 69389-倒置字符串

<https://www.nowcoder.com/practice/ee5de2e7c45a46a090c1ced2fdc62355?tpId=85&&tqId=29867&rp=1&ru=/activity/oj&qr=/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;
}

```