

# C++方向编程题答案

## 第一周

### day3

题目ID: 69385--字符串中找出连续最长的数字串

链接: <https://www.nowcoder.com/practice/bd891093881d4ddf9e56e7cc8416562d?tpId=85&&tqId=29864&rp=1&ru=/activity/oj&qu=/ta/2017test/question-ranking>

【题目解析】：

本题是一个很简单的题目，这里就不解析了。

【解题思路】：

遍历字符串，使用cur去记录连续的数字串，如果遇到不是数字字符，则表示一个连续的数字串结束了，则将数字串跟之前的数字串比较，如果更长，则更新更长的数字串更新到res。

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str,res,cur;
    cin>>str;
    for(int i=0;i<=str.length();i++)
    {
        // 数字+=到cur
        if(str[i]>='0' && str[i]<='9')
        {
            cur+=str[i];
        }
        else
        {
            // 找出更长的字符串，则更新字符串
            if(res.size() < cur.size())
                res=cur;
            else
                cur.clear();
        }
    }
    cout<<res;
    return 0;
}
```

69387-n个数里出现次数大于等于n/2的数

<https://www.nowcoder.com/practice/eac8c671a0c345b38aa0c07aba40097b?tpId=85&&tqId=29866&rp=1&ru=/activity/oj&qu=/ta/2017test/question-ranking>

### 【题目解析】：

本题题意很简单，需要找出超过一半的那个数字。需要注意这个题是一个往年面试的热门题型

### 【解题思路1】：

最简单的思路就是排序，然后找中间那个数。但是这个思路在面试是很难让面试官满意的，因为他不是最优的解法。

```
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
int main()
{
    int n;
    vector<int> vec;
    while(cin >> n)
        vec.push_back(n);

    sort(vec.begin(), vec.end());

    cout << vec[vec.size()/2-1] << endl;
    return 0;
}
```

### 【解题思路2】：

O(n)思想：因为要找过半的数，用一个变量count记录读取每个变量变化的次数，一个变量temp记录可能过半的数，先让count=1，然后让temp=vec[0]，然后往后遍历一遍，碰到和temp相同的数就给count++，否则就count--，如果，count变成0，就让temp=vec[i](vec数组遍历过程中的当前值)并让count=1，如此遍历一遍，因为有一个数过半，所以temp最后肯定存储的是过半的数

```
#include <vector>
#include <iostream>
using namespace std;
int main()
{
    int n;
    vector<int> vec;
    while(cin >> n)
        vec.push_back(n);

    int count = 1;
    int temp = vec[0];
    for(int i = 1; i < vec.size(); ++i){
        if(vec[i] == temp)
            count++;
        else
            count--;

        if(count == 0)
            temp = vec[i];
    }
    cout << temp << endl;
    return 0;
}
```

```
    {  
        temp = vec[i];  
        count++;  
    }  
}  
  
cout << temp << endl;  
return 0;  
}
```

比特科技制作