

C++方向编程题答案

第三周

day15

题目ID: 36886-查找输入整数二进制中1的个数

链接: <https://www.nowcoder.com/practice/1b46eb4cf3fa49b9965ac3c2c1caf5ad?tpId=37&&tqId=21285&rp=1&ru=/activity/oj&qru=/ta/huawei/question-ranking>

```
/*
    n & 1: 获取最低位二进制值
    每次n右移一位, 可以获取一位的二进制值, 右移32次, n变成0, 循环终止
*/

#include<iostream>
using namespace std;

int main(){
    int n;
    while (cin >> n)
    {
        int count = 0;
        while (n)
        {
            if ((n & 1) == 1)
                count += 1;
            n >>= 1;
        }
        cout << count << endl;
    }
    return 0;
}
```

36948-手套

链接: <https://www.nowcoder.com/practice/365d5722fff640a0b6684391153e58d8?tpId=49&&tqId=29337&rp=1&ru=/activity/oj&qru=/ta/2016test/question-ranking>

```
class Gloves {
public:
    //对于非0递增序列a1,a2...an, 要想最终取值覆盖每一个种类
    //n = sum(a1...an) - a1 + 1 (也就是总数减去最小值之后加一)
    //所以对于左右手套颜色都有数量的序列, 取左边和右边的最小值即可
    int findMinimum(int n, vector<int> left, vector<int> right) {

        int left_sum = 0, left_min = INT_MAX;
        int right_sum = 0, right_min = INT_MAX;

        int sum = 0;
```

```
//遍历每一种颜色的左右手套序列
for(int i=0;i<n;i++){
    //对于有0存在的颜色手套，累加
    if(left[i]*right[i]==0)
        sum += left[i] + right[i];
    //对于左右手都有的颜色手套，执行上面的策略
    //找到最小值和总数
    else{
        left_sum += left[i];
        right_sum += right[i];
        left_min = min(left_min, left[i]);
        right_min = min(right_min, right[i]);
    }
}
//结果为有左右都有数量的手套序列的结果+有0存在的手套数+最后再加一肯定就能保证了
return sum + min(left_sum-left_min+1, right_sum-right_min+1) + 1;
}
};
```