

# 愉快的代码【7.28】

我们特意加入了题解系统，也就是说，姐姐也会重新做一遍给你们题目（包括选做），然后在第二天的题目前给出姐姐自己的代码和注释作为题解或参考

如果觉得自己的代码略为臃肿，可以参考对比一下姐姐的代码；

如果觉得姐姐的代码不如自己的优秀，也可以尽情地嘲讽姐姐～

## 7.27问题1:

```
/*
    CF1339A:Filling Diamonds
    题目翻译1：用n个菱形填充4n-2个正三角形形成的某种形状，问有多少种填法
    题目翻译2：输入T个整数，然后输出它们
    思想：这题的第一想法也许是递归/递推/dp...但是如果你能看出这是道签到题的话：比如说我们用n
    个菱形去填，观察得有且只有1个菱形是竖着放的，而它竖着放的位置有n种，其他的菱形只能躺着放。我们
    先选一个位置放竖着放的菱形，其他躺着放的方式会发现是唯一的。因此总的方式数就是n
    时间复杂度：O(T)
    你能不能在五分钟内看出这个规律呢～
*/

#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int main(){
    int i,j,t,n;
    cin >> t;
    while (t--) {
        cin >> n;
        cout << n << endl;
    }
}
```

## 7.27问题2:

```
/*
    CF1355A:Sequence with Digits
    题目翻译：给出一个数a1和一个递推式，求ak是多少
    思想：这是某次codeforces常规赛的第一题，我第一眼看下去，当场吓的我不敢参加这次比赛（因
    为比赛成绩不好会掉分），过两天看了题解之后，哇哦签到题～

```

这道题的坑点就在于吓人的数据范围 ( $k < 10^{16}$ )，单纯的模拟肯定会超时的，然后想用某种数论方式处理递推式，然后毫无处理方法。最后看了题解之后发现，当  $\text{minDigit}(an) = 0$  的时候，递推之后的新数字和原来是一样的，而且之后的数字也都是一样的，那这时就可以停止递推，直接输出结果了。分析知每次递推最多在原数字上加个两位数，那么当递推次数足够多之后（可以证明次数在1000以内），这个数的百位（或者在此之前个位或十位）就会变成0，那么这时  $\text{minDigit}(an) = 0$ ，就可以停止递推了

有的签到题就告诉你就是这么做的，但是要来个可怕的数据范围吓吓某些憨憨

时间复杂度：  $O(t)$

你能不能看出这个规律呢～

```
*/

#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int ma(ll x) {
    ll num = 0;
    while (x) {
        num = max(num, x%10);
        x /= 10;
    }
    return num;
}

int mi(ll x) {
    ll num = 9;
    while (x) {
        num = min(num, x%10);
        x /= 10;
    }
    return num;
}

int main() {
    int i, j, t, x, y;
    ll n, m;
    cin >> t;
    while (t--) {
        cin >> n >> m;
        for (i=1; i<=m; i++) {
            x = ma(n);
            y = mi(n);
            if (y == 0) break;
            n += x*y;
        }
        cout << n << endl;
    }
}
```

## 今天的题目：

知识点：模拟

从数据结构与算法考试来看

大多数编程题其实都是模拟题：直接根据题意编写

反而基本没有最短路、最小生成树这种知识点偏明显的题目

而一般模拟题都是中等偏简单的题目

所以我们暑假敲代码的基本想法也是尽可能地做出中等偏简单的题

争取以后的机考能做出60%以上

（不过看了笔试题好像树论图论那些硬知识点偏多，在快开学的时候会注重一下那部分的复习）

1、<https://www.luogu.com.cn/problem/P1042>

（大一程序设计原题

2、<https://www.luogu.com.cn/problem/P1067>

（依然是中等偏简单