

愉快的代码【8.11】

【7.27】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的题目（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～

【7.30】我们特意加入了团队系统，因为感觉到你们有点像是独立学习的样子，比如说姐姐和你们之间有交流，但是你们之间有没有交流呢姐姐就感受不到啦

所以正好在洛谷上发现了一个团队系统，我们可以在这上面布置作业呀（当然姐姐也会继续以 pdf 形式布置作业，你们也还是要以 pdf 形式交作业哈），然后你们就可以在上面看到其它妹妹们的代码呀（包括 AC 代码和还未 AC 的整个过程的代码和分数呀），觉得她们表现不够自己好的话，就可以在群里尽情地嘲笑她们呀～

然后那上面还有一个比赛功能哇，具体形式和我们平时的机考差不多，暑假差不多结束了我们也会有一次期末模拟机考的哈～

如果你们开心的话，你们也可以联合起来给姐姐布置一次平时的作业呀，或者给姐姐安排一次机考呀，你们都是团队的管理员了哈

【8.1】准备给你们留个有趣的团队大作业：给姐姐安排一次机考～

具体时间、题数、难度、知识点待定～

【8.3】经过了某些人性与道德的思考，得出了一个奇怪的想法：

“我今天把代码解决了，明天姐姐的代码还有兴趣看嘛”

那就当天放出来好啦～

同样地：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～

今天的题目：

知识点：数论

数论是什么？

数论是纯粹数学的分支之一，主要研究整数的性质。和数学有关的东西

比如说我们离散数学课学的很多基础的数学问题

很多都在平时我们敲代码的时候有体现和应用

逻辑化简、各种位运算、加法乘法原理、取余、排列组合...

素数、最大公约数、最小公倍数、还有你们可能还没学到的埃氏筛...

你们还记得多少呢～

1、<https://www.luogu.com.cn/problem/P1372>

2、<https://www.luogu.com.cn/problem/P1414>

今天的答案：

8.11问题1：

```
/*
    洛谷P1372：又是毕业季I
    思想：公约数 and 结论题。1~n选k个数，使得这k个数最大公约数最大。我们设这个最大公约数为
    x，那么贪心最优选法为x, 2x, 3x, ..., kx。
    为什么不选2x, 3x, 4x, ..., (k+1)x？如果能选这个那么肯定能选上面那个，而这种选法并不会优于
    上一种。
    那么如何确定x？显然最优的选法为  $kx \leq n$  and  $k(x+1) > n$ 
    so 最后的结果就是n/k
    时间复杂度：O(1)
*/

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

int main() {
    int n, k;
    cin >> n >> k;
    cout << n/k << endl; // n/k 满足  $kx \leq n$  and  $k(x+1) > n$ 
}
```

8.11问题2：

```
/*
    洛谷P1414：又是毕业季II
    思想：约数、公约数问题。这里每个人的数字不再是1~n啦，而是n个随意的数啦。这里依然要找k个
    数，使得它们的最大公约数最大。这里我们采用暴力枚举的方式，我们设这个最大公约数为x，从大到小枚
    举x，如果这n个数里有k个或者更多个数，它们有公约数x，那么这个x就是满足k个数的最大公约数。
    时间复杂度：O(nsqrt(n))
*/

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

int v[1000001]; // 记录约数
int main() {
```

```

int i,j,n;
cin >> n;
int a[n]; // 这里的空间可以优化为O(1)
for (i=0; i<n; i++) {
    cin >> a[i];
    for (j=1; j*j<=a[i]; j++) // 找出它的所有约数，时间复杂度O(sqrt(n))
        if (a[i]%j == 0) {
            v[j]++;
            if (j*j != a[i]) v[a[i]/j]++; // 防止完全平方数重复记录
        }
}
j = *max_element(a,a+n); // algorithm库函数：返回指向容器最大值位置的指针
for (i=1; i<=n; i++)
    for (; j>=0; j--) // 从上一个位置开始往下枚举
        if (v[j] >= i) {
            cout << j << endl;
            break; // 必须break跳到下一个k
        }
}

```

Interesting thing:

又没啦