**1.Hash killer V（hash.pas/c/cpp）**

**来源：某天突发奇想**

1-4手撸即可。时间复杂度O（1）

5-8 首先发现两个串在模lcm（mo[i]）的情况下相同时大头的程序会挂掉~~（很显然呀）~~.

我们强制第一个串表示成26进制下产生的数为0，

第二个串在26进制下产生的数为lcm(mo[i]).

我们发现lcm(1-20)<1e9

然后随便乱搞就可以了

时间复杂度O（Nlogmo[i]）

9-12 在5-8的基础上加上高精度就可以了

时间复杂度O（高精度）

~~当然出题人知道你们不会去打高精度模板的~~

13-16 高精度lcm超级萎，考虑优化

我们将mo[i]暴力分解质因数。

则lcm每一个因子的个数恰好等于模数中拥有该因子最多的数字重该因子的个数。

然后高精度一发就可以了

或者我们打一个取模函数

Gcd(ans,now)=gcd(ans%now,now)

然后就可以了呀。

时间复杂度O（Nlogmo[i]+高精度）

17-20 我们考虑优化这个过程。

我们在求出前i-1个数的lcm之后，我们将其表示成i-1个数相乘的模式。

然后我们每一次新加入一个数，我们枚举前面每一个数，将新的数字除去他们的lcm，然后我们将新的数字插入到序列中。

我们只要在最后的时候将N个数乘起来就可以了。

时间复杂度O（N^2logmo[i]+高精度）

**2.LZH loves driving（drive.pas/c/cpp）**

**来源：bzoj2837小强的形状（改）**

**Bzoj2441小W的问题（改）**

1-5大力O（N^5）

6-10我们枚举k，预处理出I,j,m,l的合法方案 O（N^3）

11-15 我们枚举I,j，用树状数组维护k的答案 O（N^2logN）

16-20 我们假设123表示第一个最矮，第二个次高，第三个最高的方案数（hint：可以1==2，但是不能2==3）

同理\*\*3表示第三个最高的方案数（1，2！=3）

显然这两个都可以用树状数组求出来

然后对于一个节点，合法答案就是\*\*3-123

O(NlogN)

**3.LZH的逃课计划（truancy.pas/c/cpp）**

**来源：一道奶牛题。**

1-6：2^n不虚

7-14：设f[i]表示前i天的最优解

转移枚举j，显然逃课越多越好

O（N^2）

15-20我们反着做。

设f[i]表示前i天，且第i天逃课的最好答案

显然可以用单调队列维护

O（N）