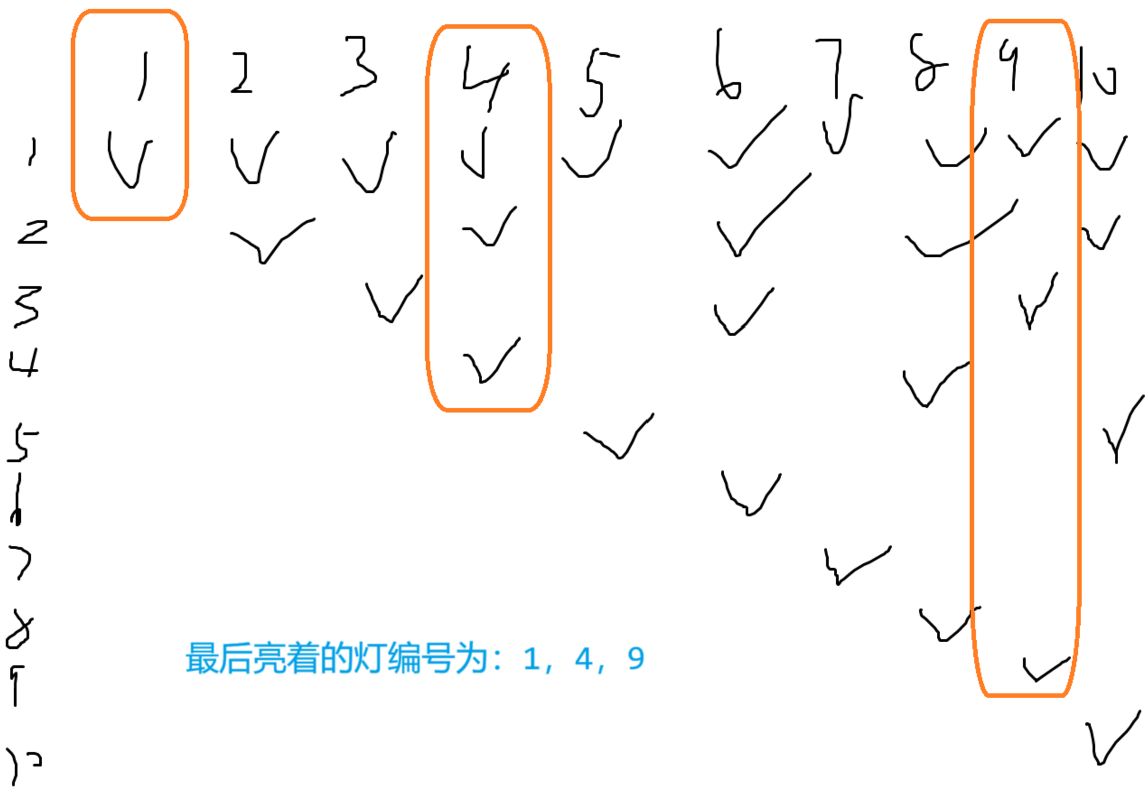
Koishi 开关灯

题意简述

有 n 盏电灯，编号 1 到 n，一开始所有灯都是关着的。有 n 次操作，第 i 次会把编号为 i 的倍数的灯的状态反转（开变关，关变开），求最后还亮着的灯的编号之和。

题目解析

首先找规律，把开关灯的情况画出来。下图是取 $n=10$ 的情况。



于是我们有个大胆的猜想：**最后亮着的灯都是完全平方数，要求的答案就是不超过 n 的完全平方数之和。**

很抱歉，这个结论真的是对的。

(不)严谨证明

注意到，每个数被操作的次数实际上是他的因数个数。

例如 6 的因数有 $1,2,3,6$，所以 6 会被操作 4 次。

而完全平方数的因数个数必然是奇数个，如 9 的因数有 $1,3,9$，9 会被操作 3 次，所以最后会开着。

所以最后亮着的灯编号必然都是完全平方数。

下面来求解不超过 n 的完全平方数之和，这里提供中学做法和竞赛做法两种做法。

中学做法

在高中的时候我们知道：

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

记 $m = \lfloor \sqrt{n} \rfloor$ ，那么直接求解 $\sum_{i=1}^{m} i^2$ 即可得到答案。

同时我们知道，乘法对于 mod 运算是不会影响结果的。但是除法会影响。

于是我们目标是消掉分母的 6。

我们知道 n 和 $n+1$ 一定一个是奇数一个是偶数，所以必然其中有一个能被 2 整除。

同理，我们知道 $n, n+1, 2n+1$ 其中必然也有一个能被 3 整除。

所以分子一定可以把分母的 6 约掉。分类讨论即可。

不可能约不掉，不然算出来的结果就不是整数了。一堆整数加起来不是整数，这不是太荒谬了吗。

代码1

```
#include<iostream>
#include<cstdio>
#include<cmath>
#define ll long long
using namespace std;

const int mod = 1e9+7;

int main(){
    long long n,q,ans;
    scanf("%lld",&n);
    q=sqrt(n);
    if(q%6==0){
        ans=((((q/6)*(q+1))%mod)*(2*q+1))%mod;
    }else if(q%3==1){
        ans=((((q*(q+1)/2))%mod)*(2*q+1)/3)%mod;
    }else{
        ans=(((q*(q+1)/6)%mod)*(2*q+1))%mod;
    }
    printf("%lld",ans);
    return 0;
}
```

竞赛做法

前面部分与中学做法相同，我们目标是把分母中的 6 搞掉。

在 mod 运算中，不能直接做除法。那有没有一种东西，把它乘上去，可以起到除以 6 的效果？

是有的。我们把这种东西称为**乘法逆元**。

在这题中，由于模数是 10^9+7，是个极大的质数，所以对于乘法逆元的求解有两种方法。

第一种方法是求解同余方程 $6x \equiv 1 \pmod{10^9+7}$，这个可以用扩展欧几里得算法（Exgcd）来求解，过程跟辗转相除法很像，感兴趣的可以自己去了解，这里不做详细赘述。

第二种方法是，由于 10^9+7 是质数，因此有费马小定理 $6^{10^9+6} \equiv 6 \pmod{10^9+7}$，变形一下就有 $6^{10^9+5} \equiv 6^{-1} \pmod{10^9+7}$，使用快速幂算法计算出 $6^{10^9+5} \mod 10^9+7$ 即可。

下面的代码采用第一种方法实现。

代码

```
#include<iostream>
#include<cmath>
#include<cstdio>
#define it long long int
#define pir pair<it,it>
using namespace std;

pir Exgcd(it a,it b){
    if(b==0) return make_pair(1,0);
    pir temp = Exgcd(b,a%b);
    return make_pair(temp.second,temp.first-a/b*temp.second);
}

it inv(it a,it b){
    pir temp = Exgcd(a,b);
    return (temp.first%b+b)%b;
}

int main(){
    it a=6,b=1e9+7,n,inv6;
    inv6=inv(a,b);
    scanf("%lld",&n);
    n=sqrt(n);
    printf("%lld",(((((n*(n+1))%b)*(2*n+1))%b)*inv6)%b);
    return 0;
}
```

补充

这题有幸被选入了第十五届蓝桥杯的校赛选拔。~~坑害了众多学长。~~

数据比较弱，有同学赛时用暴力 + 边界特判搞过去了。

~~还有直接写高精度乘除法的。好像也能过。~~