> **写在最前面：本文部分内容来自网上各大博客或是各类图书，由我个人整理，增加些许见解，仅做学习交流使用，无任何商业用途。因个人实力时间等原因，本文并非完全原创，请大家见谅。**

《算法竞赛中的初等数论》正文 0x00整除、0x10 整除相关（ACM / OI / MO）（十五万字符数论书）

《算法竞赛中的初等数论》（信奥 / 数竞 / ACM）前言、后记、目录索引（十五万字符的数论书）

全文目录索引链接： https://fanfansann.blog.csdn.net/article/details/113765056

本章内容不多，比较简单，非常基础，建议完全掌握，包括所有的概念和证明。

0x30 积性函数

一些定义

- **数论函数**
  定义域为**正整数**的函数称为数论函数。
- **积性函数**
  对于数论函数 f，若任意**互质**的 p,q 都有 $f(pq)=f(p)f(q)$，则称 f 是积性函数。
- **完全积性函数**
  对于数论函数 f，若任意 p,q 都有 $f(pq)=f(p)f(q)$，则称 f 是完全积性函数
- **定义逐点加法**
  $(f+g)(x)=f(x)+g(x)$，$(f\cdot g)(x)=f(x)g(x)$

定理30.1： 积性函数一定满足 $f(1)=1$。

考虑证明：

显然 1 与任何数都互质，满足积性函数的定义，那么我们假设存在一个正整数 a 满足 $f(a)!=0$，显然有：$f(a)=f(1\times a)=f(1)\times f(a)$，两端同除 $f(a)$ ，得：$f(1)=1$，性质得证 □

定理30.2： 对于一个大于 1的正整数 N ，根据唯一分解定理有 $N=\prod p_i^{a_i}$ ，则对于任意积性函数 f ，有： $f(N)=f(\prod p_i^{a_i})=\prod f(p_i^{a_i})$若 f完全积性，则 $f(N)=\prod f(p_i)^{a_i}$。

由此可得推论：**凡是积性函数均可用线性筛法求解**

性质30.3： 对于一个大于 1 的整数由唯一分解定理有： $n=\prod p_i^{a_i}$，其中 p_i 为互不相同的素数。

对于一个**积性函数** f，$f(n)=f(\prod p_i^{a_i})=\prod f(p_i^{a_i})$（不互质不能提出来）

对于一个**完全积性函数** f，$f(n)=\prod f(p_i)^{a_i}$

性质30.4： 若 $f(x)$ 和 $g(x)$ 均为积性函数，则以下函数也为积性函数：

$$\begin{aligned} h(x)&=f(x^p)\\ h(x)&=f^p(x)\\ h(x)&=f(x)g(x)\\ h(x)&=\sum_{d\mid x}f(d)g(\frac{x}{d}) \end{aligned} \tag{1}$$

0x31 常见积性函数

- $\varphi(n)$ - 欧拉函数 $\varphi(n)=\sum_{i=1}^{n}[i\perp n]$
- $\mu(n)$ - 莫比乌斯函数，关于非平方数的质因子数目$\mu(n)=\begin{cases}1 & n=1\\ 0 & \exists d>1,d^2\mid n\\ (-1)^{\omega(n)} & \texttt{otherwise}\end{cases}$，其中 $\omega(n)$ 表示 n 的本质不同质因子个数，它是一个加性函数。

> **加性函数**
> 此处的加性函数指数论上的加性函数 (Additive function)。对于加性函数 f，当整数 a,b 互质时，均有 $\mathrm{f}(ab)=\mathrm{f}(a)+\mathrm{f}(b)$。
> 应与代数中的加性函数 (Additive map) 区分。

- $\gcd(n,k)$ - 最大公因数，当 k 固定的情况
- 除数函数：$\sigma_k(n)=\sum_{d\mid n}d^k$
  - $\sigma_0(n)$ 通常简记作 $\mathrm{d}(n)$ 或 $\tau(n)$，
  - $\sigma_1(n)$ 通常简记作 $\sigma(n)$。
- $d(n)$ - n 的正因子数目 $d(n)=\sum_{i\mid n}1$
- $\sigma(n)$ - n 的所有正因子之和 $\sigma(n)=\sum_{d\mid n}d$
- $1(n)$ - 不变函数，定义为 $1(n)=1$ （完全积性）
- $id(n)$ - 单位函数（恒等函数），定义为 $id(n)=n$（完全积性）
- $idk(n)$ - 幂函数，对于任何复数、实数 k，定义为 $idk(n)=n^k$ （完全积性）

- $\varepsilon(n)$ - 定义为：$\varepsilon(n)=[n=1]$，若$n=1$，$\varepsilon(n)=1$；若 $n>1$，$\varepsilon(n)=0$ 。即：**对于狄利克雷卷积的乘法单位 / 狄利克雷特卷积的单位元**，（完全积性）
- $\lambda(n)$ - 刘维尔函数，关于能整除 n 的质因子的数目
- $\gamma(n)$，定义为 $\gamma(n)=(-1)^{\omega}(n)$，在此加性函数 $\omega(n)$ 是不同能整除 n 的质数的数目
- 另外，所有狄利克雷特征均是完全积性的。

积性函数 **欧拉函数** 已在 **0x14.1 欧拉函数** 中讲解过，这里不再赘述。

0x32 莫比乌斯函数

定义

$$\mu(n)=\begin{cases}0 & \exists i\in[1,m],C_i>1\\(-1)^m & \forall i\in[1,m],C_i=1\end{cases}$$

构造莫比乌斯函数

设 N 分解质因数 $N=p_1^{c_1}p_2^{c_2}\cdots p_m^{c_m}$ (是积性函数)

$$\mu(N)=\begin{cases}0 & \exists i\in[1,m],\ c_i>1\\ (-1)^m & \forall i\in[1,m],\ c_i=1\end{cases}$$

当 N 含相同质因子时，$\mu(N)=0$

当 N 的质因子各不相同

若 N 有奇数个质因子 $\mu(N)=1$

若 N 有偶数个质因子 $\mu(N)=1$

Möbus函数由来：算术函数 f，和函数 $F=\sum_{d|n}f(d)$ 很明显 F 由 f 推出.

我们可以反过来吗？也就是由 F 简便地推出 f？我们展开 $F(n)$ 来发现一些规律.

$F(1)=f(1)$，$F(2)=f(1)+f(2)$，$F(3)=f(1)+f(3)$，$F(4)=f(1)+f(2)+f(4)\cdots$

解得：$f(1)=F(1)$，$f(2)=F(2)-F(1)$，$f(3)=F(3)-F(1)$，$f(4)=F(4)-F(2)$，$f(6)=F(6)-F(3)-F(2)+F(1)$

发现 $f(n)$ 等于形式为 $\pm F(\frac{n}{d})$ 的一些项之和. 其中 $d|n$，猜测存在一个等式：

$f(n)=\sum_{d|n}\mu(d)F(\frac{n}{d})$ ($\mu(d)$ 为我们当前未知待构造的函数 也就是F的系数)

发现 $F(p)=f(1)+f(p)\Rightarrow f(p)=F(p)-F(1)$，其中 p 为质数，构造出 $\mu(p)=-1$

$F(p^2)=f(1)+f(p)+f(p^2)\Rightarrow f(p^2)=F(p^2)-(F(p)-F(1))-F(1)=F(p^2)-F(p)$

即令 $\mu(p^2)=0\Rightarrow$ 规定对任意质数 p，p^k，$k>1$，令 $\mu(p^k)=0$

若 μ 为积性函数，则 μ 的值仅由质数 p 的幂的值决定

综上所述，我们可以构造出一个全新的函数：Mfy 函数 μ.

令 $\mu(n)=\begin{cases}0, & \exists i\in[1,m],\ c_i>1\\ 1, & \forall i\in[1,m],\ c_i=1\end{cases}$ 其中 $n=p_1^{c_1}\cdot p_2^{c_2}\cdots p_m^{c_m}$

我们构造出来的Mfy函数 μ 即为莫比乌斯函数！

我们利用构造出来的 Möbus 函数 μ 实现由和函数 F 求

得 f 的过程 即为莫比乌斯反演！

我们猜测的公式 $f(n)=\sum_{d|n}\mu(d)F(\frac{n}{d})$ 即为反演公式!!!

莫比乌斯函数可以理解为一个容斥原理的映射

性质32.1：

$$\sum_{d|n}\mu(d)=\begin{cases}1 & n=1\\ 0 & n\neq 1\end{cases} \tag{2}$$

即 $\sum_{d|n}\mu(d)=\varepsilon(n)$，$\mu*1=\varepsilon$，其中 $*$ 是指 $Dirichlet$ 卷积。

> **证明**

> 设 $n=\prod_{i=1}^{k}{p_i}^{c_i}, n'=\prod_{i=1}^{k}p_i$
>
> 那么 $\sum_{d|n}\mu(d)=\sum_{d|n'}\mu(d)=\sum_{i=0}^{k}C_k^i\cdot(-1)^i=(1+(-1))^k$
>
> 根据二项式定理，易知该式子的值在 $k=0$ 即 $n=1$ 时值为 1 否则为 0，这也同时证明了 $\sum_{d|n}\mu(d)=[n=1]=\varepsilon(n)$ 以及 $\mu*1=\varepsilon$

性质32.2：

$$[\gcd(i,j)=1]=\sum_{d|\gcd(i,j)}\mu(d)$$

> **直接证明**：如果看懂了上一个结论，这个结论稍加思考便可以推出：如果 $\gcd(i,j)=1$ 的话，那么代表着我们按上个结论中枚举的那个 n 是 1，也就是式子的值是 1，反之，有一个与 $[\gcd(i,j)=1]$ 相同的值：0
>
> **利用 ε 函数**：$[\gcd(i,j)=1]=\varepsilon[\gcd(i,j)]=\mu*I=\sum_{d|\gcd(i,j)}\mu(d)$

证明 $\mu(x)$ 是积性函数

设 a,b:

$a=p_1^{\alpha_1}\times p_2^{\alpha_2}\times\cdots\times p_n^{\alpha_k}b=p_1^{\beta_1}\times p_2^{\beta_2}\times\cdots\times p_n^{\beta_t}$
$a\times b=p_1^{\alpha_1}\times p_2^{\alpha_2}\times\cdots\times p_n^{\alpha_k}\times p_1^{\beta_1}\times p_2^{\beta_2}\times\cdots\times p_n^{\beta_t}$

若 $\mu(a)=0$或 $\mu(b)=0$，则 $\mu(a\times b)=0$（因为均含有 α or $\beta>1$）

即 $\mu(a\times b)=\mu(a)\times\mu(b)$

若 $\mu(a)\neq 0$ 或 $\mu(b)\neq 0$，则 $\mu(a\times b)=(-1)^{k+t}=(-1)^k\times(-1)^t=\mu(a)\times\mu(b)$

0x33 狄利克雷卷积

- **定义：** 若函数f,g为积性函数，则定义 f,g 的狄利克雷卷积为：

$f*g(n)=\sum_{d|n}f(d)g(\frac{n}{d})$

计算的时候可以把枚举约数转换成枚举倍数，以调和级数 $O(nlogn)$ 的复杂度求出 $f*g$ 的前 n 项。详见本文 **0x33.2 $O(nlogn)$ 的卷积预处理**

- **满足性质：**

交换律：
$f*g(n)=\sum_{d|n}f(d)g(\frac{n}{d})=\sum_{d|n}g(d)f(\frac{n}{d})=g*f(n)$

结合律：

$f*g*h(n)=\sum_{d|n}f(d)\sum_{t|\frac{n}{d}}g(t)h(\frac{n}{dt})=\sum_{d_1d_2d_3=n}f(d_1)g(d_2)h(d_3)=f*(g*h)(n)$

分配律： $f*(g+h)=f*g+f*h$

等式的性质： $f=g$ 的充要条件是 $f*h=g*h$，其中数论函数 $h(x)$ 要满足 $h(1)\neq 0$。

> **证明：** 充分性是显然的。
>
> 证明必要性，我们先假设存在 x，使得 $f(x)\neq g(y)$。那么我们找到最小的 $y\in\mathbb{N}$，满足 $f(y)\neq g(y)$，并设 $r=f*h-g*h=(f-g)*h$。
>
> 则有：
>
> $$\begin{aligned} r(y) &= \sum_{d|y}(f(d)-g(d))h\left(\frac{y}{d}\right) \\ &= (f(y)-g(y))h(1) \\ &\neq 0 \end{aligned} \tag{3}$$
>
> 则 $f*h$ 和 $g*h$ 在 y 处的取值不一样，即有 $f*h\neq g*h$。矛盾，所以必要性成立。
>
> **证毕**

单位元： $f*\varepsilon=f$

逆元： 对于任何一个满足 $f(x)\neq 0$ 的数论函数，如果有另一个数论函数 $g(x)$ 满足 $f*g=\varepsilon$，则称 $g(x)$ 是 $f(x)$ 的逆元。由 **等式的性质** 可知，逆元是唯一的。

根据定义显然有

$$\begin{aligned} \epsilon &= g*f \\ [n=1] &= \sum_{i|n} f(i)\, g\left(\frac{n}{i}\right) \\ [n=1] &= f(1)\, g(n) + \sum_{i|n, i\neq 1} f(i)\, g\left(\frac{n}{i}\right) \end{aligned}$$

化简即可得到：

$$g(n)=\frac{[n=1]-\sum\limits_{i|n,i\neq 1} f(i)\, g(\frac{n}{i})}{f(1)}$$

重要性质： 若函数f,g为积性函数，则$f*g$也为积性函数（ `*` 为狄利克雷特卷积）

两个积性函数的 Dirichlet 卷积也是积性函数

> **证明：** 设两个积性函数为 $f(x)$ 和 $g(x)$，再记 $h=f*g$。
>
> 设 $\gcd(a,b)=1$，则：
>
> $h(a)=\sum_{d_1|a} f(d_1)g\left(\frac{a}{d_1}\right), h(b)=\sum_{d_2|b} f(d_2)g\left(\frac{b}{d_2}\right),$
>
> 所以：

> $$\begin{aligned} h(a)h(b) &= \sum_{d_1|a} f(d_1)g\left(\frac{a}{d_1}\right)\sum_{d_2|b} f(d_2)g\left(\frac{b}{d_2}\right) \\ &= \sum_{d|ab} f(d)g\left(\frac{ab}{d}\right) \\ &= h(ab) \end{aligned}$$
>
> 所以结论成立。
>
> **证毕**

积性函数的逆元也是积性函数

> **证明**：我们设 $f * g = \varepsilon$，并且不妨设 $f(1) = 1$。考虑归纳法：
>
> - 若 $nm = 1$，则 $g(nm) = g(1) = 1$，结论显然成立；
> - 若 $nm > 1(\gcd(n, m) = 1)$，假设现在对于所有的 $xy < nm(\gcd(x, y) = 1)$，都有 $g(xy) = g(x)g(y)$，所以有：
>
> $$g(nm) = -\sum_{d|nm,d\neq 1} f(d)g\left(\frac{nm}{d}\right) = -\sum_{a|n,b|m,ab\neq 1} f(ab)g\left(\frac{nm}{ab}\right) \tag{4}$$
>
> 又因为 $\frac{nm}{ab} < nm$，所以有：
>
> $$\begin{aligned} g(nm) &= -\sum_{a|n,b|m,ab\neq 1} f(ab)g\left(\frac{nm}{ab}\right) \\ &= -\sum_{a|n,b|m,ab\neq 1} f(a)f(b)g\left(\frac{n}{a}\right)g\left(\frac{m}{b}\right) \\ &= f(1)f(1)g(n)g(m) - \sum_{a|n,b|m} f(a)f(b)g\left(\frac{n}{a}\right)g\left(\frac{m}{b}\right) \\ &= g(n)g(m) - \sum_{a|n} f(a)g\left(\frac{n}{a}\right)\sum_{b|m} f(b)g\left(\frac{m}{b}\right) \\ &= g(n)g(m) - \varepsilon(n) - \varepsilon(m) \\ &= g(n)g(m) \end{aligned} \tag{5}$$
>
> 综合以上两点，结论成立。
>
> **证毕**

0x33.1 常见积性函数的卷积

性质0x33.1：$\forall f(n), e * f(n) = f(n)$

性质0x33.2：$1 * 1(n) = \sum_{d|n} 1 = d(n)$

性质0x33.3：$id * 1(n) = \sum_{d|n} d = \sigma(n)$

性质0x33.4： $\mu * 1(n) = \sum_{d|n} \mu(d) \times 1(\frac{n}{d}) = [n = 1] = \varepsilon(n)$

> **性质0x33.4证明：**
>
> 由唯一分解定理： $n = p_1^{k_1} p_2^{k_2} \ldots p_m^{k_m}$
>
> 代入上式得：
>
> $$\begin{aligned} \mu * 1(n) &= \sum_{d|n} \mu(d) \times 1(\frac{n}{d}) \\ &= \sum_{C_1=0}^{k_1} \sum_{C_2=0}^{k_2} \ldots \sum_{C_m=0}^{k_m} \mu(p_1^{C_1} p_2^{C_2} \ldots p_m^{C_m}) \end{aligned}$$
>
> 我们分析该和式的实际意义，显然当某个质因子数量 $c_i > 1$，则 μ 的值为 0。
> 这样的话，我们就可以去掉无用的枚举（$\mu = 0$），利用二项式定理证明：
>
> $$\begin{aligned} \mu * 1(n) &= \sum_{C_1=0}^{1} \sum_{C_2=0}^{1} \ldots \sum_{C_m=0}^{1} \mu(p_1^{C_1} p_2^{C_2} \ldots p_m^{c_m}) \\ &= \sum_{C_1=0}^{1} \sum_{C_2=0}^{1} \ldots \sum_{C_m=0}^{1} (-1)^{\sum_{i=1}^{m} C_i} \\ &= \sum_{i=0}^{m} (-1)^i \binom{m}{i} \\ &= [m = 0] = [n = 1] = \varepsilon(n) \end{aligned}$$

性质0x33.5： $\varphi * 1(n) = \sum_{d|n} \varphi(d) = n = id(n)$

> **性质0x33.5证明：**
>
> 将 n 分解质因数：$n = \prod_{i=1}^{k} {p_i}^{c_i}$
>
> 首先，因为 φ 是积性函数，故只要证明当 $n' = p^c$ 时 $\varphi * 1 = \sum_{d|n'} \varphi(\frac{n'}{d}) = \mathrm{id}$ 成立即可。
>
> 因为 p 是质数，于是 $d = p^0, p^1, p^2, \cdots, p^c$
>
> 易知如下过程：
>
> $$\begin{aligned} \varphi * 1 &= \sum_{d|n} \varphi(\frac{n}{d}) \\ &= \sum_{i=0}^{c} \varphi(p^i) \\ &= 1 + p^0 \cdot (p-1) + p^1 \cdot (p-1) + \cdots + p^{c-1} \cdot (p-1) \\ &= p^c \\ &= \mathrm{id} \end{aligned}$$

> 该式子两侧同时卷 μ 可得 $\varphi(n)=\sum_{d|n} d\cdot\mu(\frac{n}{d})$
> $\because \varphi=\mu*id, \epsilon=\mu*1$
> $\therefore 1*\varphi=1*\mu*id$
> $\therefore 1*\varphi=\epsilon*id=id(n)$
> $\therefore \sum_{d|n}\varphi(d)=n$

积性函数的转换关系：

$\mu \Rightarrow^{*1}\Rightarrow \varepsilon \Rightarrow^{*1}\Rightarrow 1 \Rightarrow^{*1}\Rightarrow d$

$\varphi \Rightarrow^{*1}\Rightarrow id \Rightarrow^{*1}\Rightarrow \sigma$

反方向的转换关系：

$\mu \Leftarrow^{*\mu}\Leftarrow \varepsilon \Leftarrow^{*\mu}\Leftarrow 1 \Leftarrow^{*\mu}\Leftarrow d$

$\varphi \Leftarrow^{*\mu}\Leftarrow id \Leftarrow^{*\mu}\Leftarrow \sigma$

0x33.2 $O(nlogn)$ 的卷积预处理

若已知数论函数 f,g，我么可以将枚举约数转换成枚举倍数，以调和级数 $O(nlogn)$ 的复杂度求出 $f*g$ 的前 n 项：

$h(n)=f*g(n)=\sum_{d\,|\,n}f(d)g(\frac{n}{d})$

推荐这个写法:

```
void Dirichlet(ll *f, ll *g)
{
    int h[N] = {0};
    for(int i = 1; i <= n; ++ i) {
        for(int j = i; j <= n; j += i) {
            h[j] = (h[j] + f[i] * g[j / i]) % mod;
        }
    }
    for(int i = 1; i <= n; ++ i)
        f[i] = h[i];
}
```

另一种写法:

```cpp
int f[N], g[N], h[N];
inline void init(int n) {
    for (int i = 1; i * i <= n; i++) {
        for (int j = i; i * j <= n; j++) {
            if (j == i) h[i * j] += f[i] * g[i];
            else h[i * j] += f[i] * g[j] + f[j] * g[i];
        }
    }
}
```

- **竞赛例题选讲**

Problem A. Longge的问题 ([P2303 [SDOI2012]](https://www.luogu.com.cn/problem/P2303))

Problem

现在问题来了：给定一个整数 n，你需要求出 $\sum_{i=1}^{n} \gcd(i,n)$，其中 $\gcd(i,n)$ 表示 i 和 n 的最大公因数。

$1 \le n \le 2^{32}$

Solution

设 $\gcd(i,n)=d$，则 $\gcd(\frac{i}{d},\frac{n}{d})=1$

d 显然就是 n 的因数

我们对于每个 d，要求有多少 i 使得 $\gcd(\frac{i}{d},\frac{n}{d})=1$ ，设求出有 x 个 i，那么对答案的贡献就是 $d\times x$。为什么是这些贡献？很显然，这些 i 与 n 的 gcd 就是 d，共 x 个这样的 i，所以是 $d\times x$ 。

因为满足 $gcd(\frac{i}{d},\frac{n}{d})=1$ 的 $\frac{i}{d}$ 的个数就是与 $\frac{n}{d}$ 互质的数，每个 $\frac{i}{d}$ 又对应一个 i ，个数就是 $\varphi(\frac{n}{d})$。前面说了 d 是 n 的因数，我们枚举所有因数，累加 $\varphi(\frac{n}{d})$ 即可。

对于每个 $\varphi(\frac{n}{d})$，$\sqrt{n}$ 求即可

当然利用欧拉反演可以直接得到一个一模一样的公式：

$$
\begin{aligned}
\sum_{i=1}^{n}\gcd(i,n) &= \sum_{i=1}^{n}\sum_{d\mid i}\sum_{d\mid n}\varphi(d)\\
&= \sum_{d\mid n}\sum_{i=1}^{n}\sum_{d\mid i}\varphi(d)\\
&= \sum_{d\mid n}\left\lfloor\frac{n}{d}\right\rfloor\varphi(d)
\end{aligned}
$$

Code

```cpp
#include <bits/stdc++.h>

using namespace std;
#define int long long
const int N = 50007;

int n, m, t;
int ans;

int get_phi(int n)
{
    int ans = n;
    for(int i = 2; i * i <= n; ++ i) {
        if(n % i == 0) {
            ans = ans / i * (i - 1);
            while(n % i == 0) n /= i;
        }
    }
    if(n > 1) ans = ans / n * (n - 1);
    return ans;
}

signed main()
{
    scanf("%lld", &n);
    int ans = 0;
    for(int i = 1; i * i <= n; ++ i) {
        if(n % i == 0) {
            ans += n / i * get_phi(i);
            if(i * i != n) {
                ans += i * get_phi(n / i);
            }
        }
    }
    printf("%lld\n", ans);
    return 0;
}
```

Problem B. Clarke and math（HDU 5628 ）

给定 $f(i)$，$(i = 1, 2, \ldots, n)$

求:

$$g(i)=\sum_{i_1|i}\sum_{i_2|i_1}\sum_{i_3|i_2}\cdots\sum_{i_k|i_{k-1}}f(i_k) \mod 1000000007(1\le i\le n)$$

Solution

f 和 g 进行一次Dirichlet卷积是 $h(n)=f*g(n)=\sum_{d|n}f(d)g(\frac{n}{d})$，显然可以发现题目中的式子实际上就是 $f*1^k$，其中 1 是不变函数，即 $1(n)=1$。使用快速幂加速即可。

Time

$O(nlog^2n)$

Code

```
// Problem: Clarke and math
// Contest: HDOJ
// URL: http://acm.hdu.edu.cn/showproblem.php?pid=5628
// Memory Limit: 65 MB
// Time Limit: 5000 ms
//
// Powered by CP Editor (https://cpeditor.org)

#include <bits/stdc++.h>

using namespace std;
using ll = long long;
const int N = 5e5 + 7, mod = 1e9 + 7;

int n, m, k;
ll f[N], yi[N], res[N];

void Dirichlet(ll *f, ll *g)
{
    int h[N] = {0};
    for(int i = 1; i <= n; ++ i) {
        for(int j = i; j <= n; j += i) {
            h[j] = (h[j] + f[i] * g[j / i]) % mod;
        }
    }
    for(int i = 1; i <= n; ++ i)
        f[i] = h[i];
}

void fpow(ll *res, ll *yi, int k)
{
    while(k) {
        if(k & 1) Dirichlet(res, yi);
```

```
        Dirichlet(yi, yi);
        k >>= 1;
    }
}

void solve()
{
    scanf("%d%d", &n, &k);
    for(int i = 1; i <= n; ++ i) {
        scanf("%lld", &f[i]);
        res[i] = 0;
        yi[i] = 1;
    }
    res[1] = 1;
    fpow(res, yi, k);
    Dirichlet(f, res);

    for(int i = 1; i <= n; ++ i) {
        printf("%lld%s", f[i], i == n ? "\n" : " ");
    }
}

int main()
{
    int t;
    scanf("%d", &t);
    while(t -- ) {
        solve();
    }
    return 0;
}
```

Problem C.

建议阅读的拓展内容：[浅谈一类积性函数的前缀和]