数论

快速幂

```
ll qpow(ll a,ll k,ll p)
{
    ll res = 1;
    for(;k;k>>=1,a=a*a%p)
        if(k&1) res=res*a%p;
    return res;
}
```

扩展欧几里得

解 $ax+by=\gcd(a,b)$ 的一组可行解。解保证 $|x|\le b,\ |y|\le a$

```
ll exgcd(ll a,ll b,ll& x,ll& y)
{
    if(!b) { x=1; y=0; return a; }
    ll d = exgcd(b, a%b, y, x);
    y -= (a/b)*x;
    return d;
}
```

乘法逆元

```
ll inv(ll a, int p)
{
    ll x,y;
    ll d = exgcd(a,p,x,y);
    return d==1?(x+p)%p:-1;
}
```

$\Theta(n)$ 求 $1\sim n$ 的逆元

```
int n=1e6, p=998244353;
vector<ll> res(n+1);
res[1] = 1;
for(int i=2;i<=n;i++) res[i] = (ll)(p-p/i)*res[p%i]%p; // 结果: res
```

$\Theta(n)$ 求任意 n 个数的逆元

```
vector<int> v;  // 所给 n 个数
int n=v.size(), p=998244353;
vector<ll> s(n+1), sv(n+1), res(n);
s[0] = 1;
for(int i=1;i<=n;i++) s[i] = s[i-1]*v[i-1]%p;
sv[n] = inv(s[n], p);
for(int i=n;i>=1;i--) sv[i-1] = sv[i]*v[i-1]%p;
for(int i=0;i<n;i++) res[i] = sv[i+1]*s[i]%p; // 结果: res
```

欧拉函数

$\varphi(x)$ 代表小于等于 x 的与 x 互质的数的个数，$\varphi(1)=1$

```
ll phi(ll x)
{
    ll res = x;
    for(ll i=2;i*i<=x;i++)
    {
        if(x%i) continue;
        res = res/i*(i-1);
        while(x%i==0) x /= i;
    }
    if(x>1) res = res/x*(x-1);
    return res;
}
```

欧拉定理/扩展欧拉定理

$$\gcd(a,m)=1 \to a^{\varphi(m)} \equiv 1 \pmod m$$

$$a^b \equiv \begin{cases} a^{b \bmod \varphi(m)}, & \gcd(a,m)=1, \\ a^b, & \gcd(a,m)\neq 1, b<\varphi(m), \\ a^{(b \bmod \varphi(m))+\varphi(m)}, & \gcd(a,m)\neq 1, b\geq\varphi(m). \end{cases} \pmod m$$

扩展欧拉定理求 $a^k \bmod p$，k 非常大

```
ll exeuler(ll a, const string& k, int p)
{
    ll phip = phi(p);
    ll t = 0;
    bool flag = false;
    for(auto c : k)
    {
        t = t*10+c-'0';
        if(t>phip)
        {
            t %= phip;
            flag = true;
        }
    }
    if(flag) t += phip;
    return qpow(a, t, p);
}
```

线性筛

```
int n = 1e6;
vector<bool> vis(n+1);
vector<int> pri;    // n 以内质数
vector<int> phi(n+1);   // 1~n 欧拉函数
phi[1] = 1;
for(int i=2;i<=n;i++)
{
    if(!vis[i])
```

```
    {
        pri.push_back(i);
        phi[i] = i-1;
    }
    for(auto j : pri)
    {
        if((ll)i*j>n) break;
        vis[i*j] = true;
        if(i%j==0)
        {
            phi[i*j] = phi[i]*j;
            break;
        }
        else phi[i*j] = phi[i]*(j-1);
    }
}
```

中国剩余定理

$$\begin{cases} x & \equiv a_1 \pmod{r_1} \\ x & \equiv a_2 \pmod{r_2} \\ & \vdots \\ x & \equiv a_k \pmod{r_k} \end{cases}$$

保证模数 r 两两互质，求解 x

```
ll crt(const vector<ll>& a, const vector<ll>& r)
{
    int k = a.size();
    ll n=1, ans=0;
    for(int i=0;i<k;i++) n = n*r[i];
    for(int i=0;i<k;i++)
    {
        ll m = n/r[i], b, y;
        exgcd(m, r[i], b, y);
        ans = (ans+a[i]*m*b%n)%n;
    }
    return (ans%n+n)%n;
}
```

数论分块

$$\sum_{i=1}^{n} f(i) \left\lfloor \frac{n}{i} \right\rfloor$$

```
ll sum = 0;
for(ll l=1,r;l<=n;l=r+1)
{
    r = n/(n/l);
    // 显然若没有 f(i) 即 f(i)=1，把 pre[r]-pre[l-1] 替换为 r-l+1
    sum += (pre[r]-pre[l-1])*(n/l);
}
cout<<sum<<endl;
```