$$Designed\ By\ FrankWkd-\ Luogu@Lwj54joy, uid=845400$$

引言：本文LateX较多,可能加载卡顿,请耐心等待.

关于Euler函数

- $Euler$ 函数，同时被称为欧拉函数，**定义：** $\phi(n)$ 表示 $\le n$ 且与 n 互质的数的个数
- 我们规定：$\phi(1)=1$

Euler函数的性质

- 积性：对于 $gcd(a,b)=1$ 的两个数 a,b，$\phi(a\times b)=\phi(a)\times\phi(b)$.
- 欧拉反演：$\Sigma_{d\mid n}\phi(d)=n$
- 对于任意质数 p，$\phi(p^k)=p^k-p^{k-1}$

对于Euler函数性质的证明

- 1：根据积性函数的定义可得。
- 2：公式解释：对于每个是 n 的质因子的数 p，把他们的 ϕ 全部加在一起的和是 n. 拿 10 举例：10 的全部质因子为 $\{2,3,5,7,9\}$,他们的 ϕ分别为: $\{0,1,2,3,4\}$, 加起来就是原数 10.
- 3: 很显然,小于 p^k 的数中与 p 不互质的数仅有$p,2p,3p\dots p^{k-1}p$.总数为 p^{k-1} 个. 那小于它且与它互质的数的个数就是总数减去 p^{k-1} 个,即 $\phi(p^k)=$ 总数

$-p^{k-1}$,即: $\phi(p^k)=p^k-p^{k-1}$

求Euler函数

单个欧拉函数求法

- 可以直接求出其唯一分解式然后求解(本人没太听懂,但是没关系,这一点用不到)

线性筛求Euler函数

- 本文的重头戏来了!
- 我们可以在线性筛求质数的同时求出每个数的Euler函数.**当 p 为 i 的最小质因子时**,有三种情况:
    - 1: i 是质数: $\phi(i)=i-1$
    - 2:p 是 $\frac{i}{p}$ 的质因子:$\phi(i)=p\times\phi(\frac{i}{p})$
    - 3: p 和$\frac{i}{p}$互质:$\phi(i)=\phi(p)\times\phi(\frac{i}{p})$.
- 为什么要解决 $\frac{i}{p}$ 与 p 的关系呢? 因为他们的乘积就是 i ,我们要求 $\phi(i)$,那么根据 $Euler$ 函数的性质 1 ,可以得出: $\phi(i)=\phi(\frac{i}{p})\times\phi(p)$,那么问题就变成了计算两个已知质数的 ϕ 的乘积.详见 板块【由数学到程序】

线性筛求Euler函数三种形式的证明

- 1: 因为 i 为质数,由定义立得:$\phi(i)=i-1$
- 2: 设 $i=p^kq$,其中 q 不含质因子 p. 利用积性可得: $\phi(i)=\phi(p^k)\times\phi(q)$ 利用 $Euler$函数性质三可得: $\phi(i)=\phi(p^k)\times\phi(q)=(p^k-p^{k-1})\times\phi(q)$ 提出公因式 p 可得: $\phi(i)=\phi(p^k)\times\phi(q)=(p^k-p^{k-1})\times\phi(q)=p(p^{k-1}-p^{k-2})\times\phi(q)$ 由 $Euler$ 函数性质三的逆运算可得: $p(p^{k-1}-p^{k-2})\times\phi(q)=p\times\phi(p^{k-1})\times\phi(q)$ 由 $Euler$ 函数性质一的逆运算可得: $p(p^{k-1}-p^{k-2})\times\phi(q)=p\phi(p^{k-1})\times\phi(q)=p\times\phi(p^{k-1}\times q)$ 根据除法的性质可得: $p\times\phi(p^{k-1}\times q)=p\times\phi(\frac{i}{p})$ 又因为 $\phi(i)=p\times\phi(p^{k-1}\times q)$ 所以得出: $\phi(i)=p\times\phi(\frac{i}{p})$
- 3: 因为这二者互质,有积性立得: $\phi(i)=\phi(p)\times\phi(\frac{i}{p})$. OK，我们已经推完了数学部分!

由数学到程序

那么，如何应用到程序中呢?

- 有几个条件:
    - 不同的量符合其性质（互质相等因数、大小关系等）
    - 等式成立。
- 第一种情况非常简单，若 i 为质数，只要将他的 ϕ 设为 $i-1$ 即可。在线性筛筛出质数时将它的 ϕ 设为其减一即可。
- 第二、三种情况需要构造两个**积为** i 的量，我们令他们分别为：p，$\frac{i}{p}$ 。因为 p 必须是 i 的最小质因子，所以它肯定是 $primes$ 数组中的值，而且 i 由两个数的乘积构成，所以是合数。那线性筛中哪里可以筛除合数呢？双层 for 中！那么，我们已经确定了大致位置。
- 情况二：

线性筛筛Euler函数代码(还没写完)

▶ 点击查看代码

```
#include <bits/stdc++.h>
int primes[101000], n, phi[101000]; //分别存储:素数,数据总数,i的欧拉函数
bool f[101000];//标记当前数是不是素数
void get_phi() {//设p是i的最小质因子，p为质数
        int cnt = 0;
        phi[1] = 1;
        for (int i = 2; i <= n; i++) {
                if (!f[i]) {
                        primes[++cnt] = i;
                        phi[i] = i - 1; //对应情况一
                }
                for (int j = 1; j <= cnt and primes[j]*i <= n; j++) {
                        f[i * primes[j]] = true;


                        /*在此for循环中，i的定义发生了转移，由var_i*primes[j]来表
示i,

                        那么var_i与primes[j]就理所当然地成为了i的因子。
                        由于primes[j]不可能比var_i更大(因为大于i的质数还没筛到)s
                        所以primes[j]符合公式中p的定义。
                        而var_i呢？它很被动，必须保证与p的乘积为i，所以干脆把var_i在
数学公式中定义为i/p

                        */
                        if (i % primes[j] == 0) {
                                phi[i * primes[j]] = phi[i] * primes[j];
                                break;
```

```
                        }else{
                                phi[i * primes[j]] = phi[i] *
phi[primes[j]];
                        }
                }
        }
}

int main() {
        std::cin >> n;
        get_phi();
        for(int i = 1;i <= n;i++){
                std::cout<<phi[i]<<std::endl;
        }
}
```