欧拉函数的定义：

$ \phi(x) $ 代表 在1 ~ n 中， 与 n 互质的数的个数

欧拉函数的性质：

- 1. 若p是质数，则$\phi(p)=p-1$
- 2. 若p是质数，则$\phi(p^k)=(p-1)p^{k-1}$
- 3. 积性函数：若$gcd(m,n)=1$，则$\phi(mn)=\phi(m)\phi(n)$

欧拉函数的计算公式

由唯一分解定理得 $$ n= \prod ^ s_{i=1} p_i ^ {\alpha_i} = p_1 ^ {\alpha_1} p_2 ^ {\alpha_2} p_3 ^ {\alpha_3}... $$
得到 $$ \phi(x)=n\times\prod^s_{i=1} \frac{p_i-1}{p_i}=n\times\frac{p_1-

1}{p_1}\times\frac{p_2-1}{p_2}\times...\times\frac{p_s-1}{p_s} $$

方法一：利用容斥原理：对于 $\forall n \in \mathbb{N}_+$

- 假设 n 只存在质因子 p, q ，

则与 n 互质的数的集合需要除去 $p, 2p, 3p, \cdots, \lfloor \frac{N}{p} \rfloor p$ 以及 $q, 2q, \cdots, \lfloor \frac{N}{q} \rfloor q$

根据容斥原理，需要补回 pq 的倍数 $pq, 2pq, \cdots, \lfloor \frac{N}{pq} \rfloor pq$

即 $\phi(n) = n - \frac{n}{p} - \frac{n}{q} + \frac{n}{pq} = n(1 - \frac{1}{p})(1 - \frac{1}{q})$

- 同理，根据容斥原理，若 n 有若干质因子 $p1, p2, \cdots, p_n$ ，则 $\phi(x) = x(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_n})$
- 再证函数的积性：

设互质的两个正整数 n, m 分别有质因子 $p_1, p_2, \cdots, p_n$ 和 $q_1, q_2, \cdots, q_m$

则 nm 的质因子为 $p_1, p_2, \cdots, p_n, q_1, q_2, \cdots, q_m$

$$\therefore \phi(nm) = nm(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_n})(1 - \frac{1}{q_1})(1 - \frac{1}{q_2}) \cdots (1 - \frac{1}{q_m})$$
$$= n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_n}) m(1 - \frac{1}{q_1})(1 - \frac{1}{q_2}) \cdots (1 - \frac{1}{q_m})$$
$$= \phi(n)\phi(m)$$

欧拉函数的计算方法：

- 1. 对于单个数的欧拉函数的求法: 设$x_1, x_2, x_3, ...$ 为n的质因子， 按照公式 $$ \phi(n) = n \times \frac{x_1-1}{x_1} \times \frac{x_2-1}{x_2} ... $$ 即可求得

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

- 2. 对于1 ~ n内所有数字的欧拉函数的求法：
  (类似素数筛法)

我们知道在线性筛中，每个合数都是被它最小的质因子筛掉的。
我们可以假设p_j是m的最小质因子，则m可以通过$p_j\times i$筛掉。 由此我们可以分类讨论：

> 若i能被p_j整除,则表示i包含了m的所有质因子 $$ \phi(m)=m\times \prod^s_{k=1}\frac{p_k-1}{p_k}=p_j\times i \times \prod^s_{k=1}\frac{p_k-1}{p_k}=p_j\times \phi(i) $$
> 若i不能被p_j整除,则i和p_j是互质的 $$ \phi(m)=\phi(p_j\times i)=\phi(p_j)\times \phi(i)=(p_j-1)\times\phi(i) $$

由此得出代码

```
std::vector<int>phi(n + 1), prime;
std::vector<bool>vis(n + 1);

phi[1] = 1;

for(int i = 2; i <= n; ++i){
    if(!vis[i]){
        prime.push_back(i);
        phi[i] = i - 1;
    }
    for(int j = 0; i * prime[j] <= n; ++j){
        int k = i * prime[j];
        vis[k] = 1;
        if(i % prime[j] == 0){
            phi[k] = prime[j] * phi[i];
            break;
        }else phi[k] = (prime[j] - 1) * phi[i];
    }
}
//其中phi数组即为每一位的欧拉函数值
```