

质数筛

线性筛, 复杂度 $O(N)$

```
const int N = 1e8 + 5;
int primes[N], tot;
bool vis[N];

void get_primes(int n){
    for (int i = 2; i <= n; i ++ ){
        if (!vis[i]) primes[tot++] = i;
        for (int j = 0; primes[j] <= n / i; j ++ ){
            vis[primes[j] * i] = true;
            if (i % primes[j] == 0) break;
        }
    }
}
```

求欧拉函数

$$x = p_1^{a_1} * \dots * p_n^{a_n}$$

$$\phi(x) = x * (1 - \frac{1}{p_1}) * \dots * (1 - \frac{1}{p_n})$$

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

筛法求欧拉函数

```
int euler[N], primes[N], tot;
bool vis[N];
void get_eulers(int n){
    euler[1] = 1;
    for (int i = 2; i <= n; i ++ ){
        if (!vis[i]){
            primes[tot ++ ] = i;
            euler[i] = i - 1;
        }
    }
}
```

```
    }  
    for (int j = 0; primes[j] <= n / i; j ++ ){  
        int t = primes[j] * i;  
        vis[t] = true;  
        if (i % primes[j] == 0){  
            euler[t] = euler[i] * primes[j];  
            break;  
        }  
        euler[t] = euler[i] * (primes[j] - 1);  
    }  
}  
}
```