Matemática

Vetor de primos até N

```
const int MAX = 50000; // Números primos até MAX
vector<int> primes(){
    bitset<MAX> sieve;
    vector<int> ps;

    sieve.set();            // Todos são "potencialmente" primos
    sieve[1] = false;       // 1 não é primo

    for (int i = 2; i <= MAX; ++i){
        if (sieve[i]){
            // i é primo
            ps.push_back(i);

            for (int j = 2 * i; j <= N; j += i)
                sieve[j] = false;
        }
    }

    return ps;
}
```

Maior Divisor Comum

Dados dois inteiros a e b, o maior divisor comum (MDC) de a e b (notamos d = (a, b) é o inteiro não-negativo d tal que d divide a e d divide b; se c divide a e c divide b, então c divide d.

```
long long gdc(long long a, long long b){
    long long rest;
    do{
        rest = a%b;
        a=b;
        b=rest;
    }while(rest!=0);
    return a;
}
```

Complexidade O(log a +log b)

Menor Múltiplo Comum

Sejam a e b dois inteiros. O menor múltiplo comum (MMC) de a e b (notamos m = [a,b]) é o inteiro m tal que a divide m e b divide m;

```
long long lcm(long long a, long long b){
    return (a/gcd(a, b))*b;
}
```

Complexidade O(log a + log b)

Veja que, na implementação acima, a divisão é feita antes do produto: esta ordem pode evitar overflow em alguns casos.

Número de Divisores

A fatoração de um número n também permite computar o número de divisores deste número: basta fazer o produto de todos os expoentes da fatoração, somados cada um de uma unidade. Veja o código abaixo.

```
long long number_of_divisors(long long n)
{
    long long res = 0;

    for (long long i=1; i*i <= n; ++i)
    {
        if (n%i == 0)
            res += (i == n/i ? 1 : 2);
    }

    return res;
}
```

Complexidade O(sqrt n)