Sieve of Eratosthenes

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

2 -> 4
3 -> 9
5 -> 25
7 -> 49

```
vector<bool> isprime(N, true);
    isprime[0] = isprime[1] = false;

    for(int i=2;i*i<N;i++){
        if(!isprime[i]) continue;
        for(int j=i*i;j<N;j+=i){
            isprime[j]=false;
        }
    }

    Time Complexity - O(nloglogn)
```

Q. Calculate the smallest prime factor of every number in the range 1 to N (N<=1e6).

```
const int N = 1e7+5;

vector<int> spf(N, 0);

for(int i=2;i*i<N;i++){
    if(spf[i]!=0) continue;
    spf[i]=i;
    for(int j=i*i;j<N;j+=i){
        if(spf[j]==0) spf[j]=i;
    }
}
```

60 -> 2
30 -> 2
15 -> 3
5 -> 5
1

Q. Calculate the number of divisors of every number in the range 1 to N (N<=1e6).

```
const int N = 1e7+5;

vector<int> divisors(N, 0);

for(int i=1;i<N;i++){
    for(int j=i;j<N;j+=i){
        divisors[j]++;
    }
}
```

```
(N + N/2 + N/3 + N/4 + ... + 1) => N * (1 + 1/2 + 1/3 + ... + 1/N) =>
O(NlogN)
```

Q. https://codeforces.com/contest/1594/problem/C

Theofanis has a string $s_1 s_2 \ldots s_n$ and a character c. He wants to make all characters of the string equal to c using the minimum number of operations.

In one operation he can choose a number x $(1 \le x \le n)$ and **for every position** i, where i is **not** divisible by x, replace s_i with c.

Find the minimum number of operations required to make all the characters equal to c and the x-s that he should use in his operations.

First Approach -> Check from N to N/2+1

```
for(int i=n;i>n/2;i--){
     if(s[i-1]==c) ans = i;
}
```

Second Approach -> If for some i all its multiples are c then the answer is 1 operation with that i.

```
for(int i=1;i<=n;i++){
     bool ok = 1;
     for(int j=i;j<=n;j+=i){
          if(s[j-1]!=c) ok=0;
     }
     if(ok) ans = i;
}
```

Segmented Sieve

```
Q. find number of primes in range l to r
given,

r - l <= 10^5
r <= 10^12
```

```
int l = 1e9, r = 1e9 + 10;
    int n = sqrtl(r) + 2;

    // find all primes <= n
```

```cpp
    vector<int> primes;
    vector<bool> isprime(n, true);

    for (int i = 2; i * i < n; i++) {
        if (!isprime[i])
            continue;
        primes.push_back((i));
        for (int j = i * i; j < n; j += i) {
            isprime[j] = 0;
        }
    }
    // nlog log n

    // l          -> 0
    // l + 1      -> 1
    // r          -> r-l+1

    // size = r - l + 1;

    // l = 99789
    // p = 7
    // first multiple of 7 after l = ceil(99789/7) * 7

    // floor(x/y) = x/y
    // ceil(x/y) = (x+y-1) / y

    vector<int> isprimeSeg(r - l + 1, true);

    for (int p : primes) {
        for (int i = max(p * p, ((l + p - 1) / p) * p); i <= r; i += p) {
            isprimeSeg[i - l] = false;
        }
    }
    // (r-l) * log log (r)

    for (int i = 0; i < r - l + 1; i++) {
        if (isprimeSeg[i]) {
            cout << i + l << " ";
        }
    }
// tc : O((R-L+1)loglog(R)+√RloglogR).
```

2nd approach:

```
// app: 2
    for (int p = 2; p <= sqrt(r); p++) {
        for (int i = max(p * p, ((l + p - 1) / p) * p); i <= r; i += p) {
            isprimeSeg[i - l] = false;
        }
    }

    n/2 + n/3 + ... + n/sqrt(r)

    n(1/2 + 1/3 + .... + 1/sqrt(r)) n log(sqrt(r)) = n log(r)

    tc = O(n*logr + sqrt(r))
```

Fermat Little theorem

$$a^{p-1} \equiv 1 \bmod p$$

where p is prime.

$a^b \bmod p$.

$\equiv a^{(p-1)x+t} \bmod p$

where $b = (p-1)x + t$

$t = b \bmod (p-1)$

$\equiv (a^{(p-1)x}) \cdot (a^t) \bmod p$

$\equiv (a^{(p-1)})^x \cdot a^t \bmod p$

$\equiv 1^x \cdot a^t \bmod p$.

$\equiv a^t \bmod p$

$\equiv a^{b \bmod (p-1)} \bmod p$

b=58 p-1=7

7 * floor(58/7) + 58%7
7 * 8 + 2

13

4 * (13/4) + 13%4
P-1 x t
A, b

G
Proof:
Let two numbers be a and b and their gcd be d.
Then we have a = dk1 and b = dk2 where k1 and k2 are integers.
Now let a > b, Then according to Euclid Division lemma we have
a = bK3 + x
Now substituting,
d*k1 = d*k2 * k3 + x
=> x = d*(k1 - k2*k3)
So d is also a factor of x.
so a%b = x have same factor d and it will be the largest common with b = d*k2
Hence gcd(a, b) = gcd(a%b, b)

10 24

10 24%10

10 4

2 4

2 0

```
int gcd (int a, int b) {
    if (b == 0)
        return a;
    else
        return gcd (b, a % b);
}
```

(A^(b))%mod

b%(mod - 1)

((a^(b^(c^d))))

Goldman Conjencture

Every even number greater than 2 can be represented as a sum of 2 prime numbers

https://codeforces.com/problemset/problem/584/D