Prime numbers

The positive integer n is called **prime** if it has only two divisors: 1 and n.

Representation of the number n in the form $p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$, where p_i are primes, is called **factorization**.

The **fundamental theorem of arithmetic** states that every integer greater than 1 either is a prime number itself or can be represented as the product of prime numbers and moreover, this representation is unique, up to the order of the factors.

For example, $12 = 2^2 * 3$, $100 = 2^2 * 5^2$.

Theorem. Number 1 is neither prime nor composite.

Proof. Number 1 is not composite because it does not have any divisor except itself. If 1 is prime, then for example number 6 has more than one factorization:

$$6 = 2 * 3,$$
$$6 = 1 * 2 * 3,$$

which contradicts the fundamental theorem of arithmetic.

Euclid theorem. The number of primes is infinite.

Proof. Let we have only n primes: $p_1, p_2, \dots, p_n$. But the number $N = p_1 p_2 * \dots * p_n + 1$ is not divisible by any of p_i $(1 \le i \le n)$ and thus is not composite. We have a contradiction.

Let $\pi(n)$ be the number of prime numbers, less or equal to n. For example,

$$\pi(1) = 0,\ \pi(2) = 1,\ \pi(20) = 8,\ \pi(10^7) = 664579$$

The value of $\pi(n)$ asymptotically equals to

$$\pi(n) \approx n / \log(n)$$

Consequence. The probability of a random number x in the range from 1 to n to be prime is

$$P(x \text{ is prime}, 1 \le x \le n) = (n / \log(n)) / n = 1 / \log(n)$$

Dirichlet's theorem on arithmetic progression. Any arithmetic progression $a + kd$ $(k = 0, 1, 2, \dots)$, where k and a are coprime, contains infinitely many prime numbers.

E-OLYMP **1616. Prime number?** Check if the given number is prime. The number is prime if it has no more than two divisors: 1 and the number itself.

► The number is called ***prime*** if its only factors are 1 and itself.

Theorem. If number n is composite, then it has a divisor no more than $\lfloor \sqrt{n} \rfloor$.

Proof. Let n be composite and d its divisor. Then n / d is also a divisor of n. Assuming that all divisors of n are greater than $\lfloor\sqrt{n}\rfloor$, then $d > \lfloor\sqrt{n}\rfloor$ and $n / d > \lfloor\sqrt{n}\rfloor$. Hence we have $d * (n / d) > \lfloor\sqrt{n}\rfloor * \lfloor\sqrt{n}\rfloor$ or $n > n$. Contradiction.

To test the number n for primality, it is enough to check whether it is divisible by one of the numbers from 2 to $\lfloor\sqrt{n}\rfloor$ inclusive. If n is divisible by at least one of them, then n is ***composite***. Otherwise it is ***prime***.

Function ***IsPrime*** returns 1 if number n is prime and 0 if n is composite.

```
int IsPrime(int n)
{
  for(int i = 2; i <= sqrt(n); i++)
    if (n % i == 0) return 0;
  return 1;
}
```

E-OLYMP [1642. Hometask](#) Kolya is still trying to pass a test on Numbers Theory. The lecturer is so desperate about Kolya's knowledge that she gives him the same task every time.

The problem is to check if $n!$ is divisible by n^2.

► Since $n! = 1 * 2 * \ldots * n$, then $n!$ Is divisible by n. If n is prime, the product $n! / n = 1 * 2 * \ldots * (n - 1)$ is not divisible by n. Therefore, for a prime n, the value of $n!$ is not divisible by n^2.

If n is not prime, it can be represented as a product of two numbers (not necessarily prime): for example, $n = a * b$. Then $n! = (a * b)!$ contains factors a, b, $a * b$ and $n!$ is divisible by n^2.

Consider two cases separately:

- for $n = 1$ (neither prime nor composite) the value of $n!$ is divisible by n^2;
- for $n = 4$ (composite) the value of $n!$ is not divisible by n^2.

Consider the examples:

If $n = 5$ (prime), then $5! = 1 * 2 * 3 * 4 * 5$ is not divisible by 5^2.

If $n = 15$ (composite), then 15! In its factorization contains the multiples 3, 5 and 15, and therefore is divisible by 15^2.

If $n = 4$ (composite), then $4! = 1 * 2 * 3 * 4$ is divisible by 4, but is not divisible by 4^2.

E-OLYMP [572. The lesson of mathematics](#) Factorize the positive integer n. For example, 3240 you must represent in the form 2^3*3^4*5.

► In the problem you need to factorize the number n. To do this, sort all its prime divisors from 2 to $\sqrt{n}$, and for each divisor count the number of its occurrences in n.

The function ***factor*** factorize the number n.

```
void factor(int n)
{
```

```
  for(int i = 2; i <= sqrt(n); i++)
  {
    int c = 0;
    if (n % i) continue;
```

Number i is a divisor of n. In the variable c compute the exponent with which it is included in the factorization of n.

```
    while(n % i == 0) n /= i, c++;
```

Print the multiple i^c. If $c = 1$, print only the value of i.

```
    if (c > 1) printf("%d^%d",i,c); else printf("%d",i);
```

If the number n is not factored yet ($n > 1$), print the multiplication sign.

```
    if (n > 1) printf("*");
  }
```

If after the end of the loop the value of n is greater than 1, then it is prime.

```
  if (n > 1) printf("%d",n);
  printf("\n");
}
```

E-OLYMP **1352. The number of primes** Vasya loves prime numbers. He decided to find a sum of first n prime numbers, that will be divisible by k. Help him.

► Iterate over the first primes (2, 3, 5, 7, ...), sum them up. As soon as the sum is divided by k, print the number of used terms.

i	2	3	4	5	6	7	8	9	10	11	12
primes[i]	1	1	0	1	0	1	0	0	0	1	0
sum	2	5		10		17				28	

The sum of the first five primes is 2 + 3 + 5 + 7 + 11 = 28, it is divisible by 7.

E-OLYMP **414. Equation** The positive integer n is given. How many solutions in positive integers have the next equation:

$$\frac{1}{n} = \frac{1}{x} + \frac{1}{y}$$

► The equation $\frac{1}{n} = \frac{1}{x} + \frac{1}{y}$ is equivalent to $\frac{1}{n} = \frac{x+y}{xy}$,

$$xy = n(x+y),\ (x-n)(y-n) = n^2$$

The number of solution pairs (x, y) equals to the number of ways to represent number n^2 as a product of two factors. This, in turn, equals to the number of divisors for the value of n^2.

If $n = p_1^{a_1} p_2^{a_2} \ldots p_k^{a_k}$, the number of its divisors equals to

$$d(n) = (a_1 + 1) * (a_2 + 1) * \ldots * (a_k + 1)$$

The answer is the value $d(n^2) = (2a_1 + 1) * (2a_2 + 1) * \ldots * (2a_k + 1)$.

For example, for $n = 2$ we have: $d(2^2) = 2 * 1 + 1 = 3$.
For $n = 12$ factorization is $12 = 2^2 * 3$. So

$$d(12^2) = (2 * 2 + 1) * (2 * 1 + 1) = 5 * 3 = 15$$

Number 12^2 for example can be represented in the form 4 * 36. Let's solve the system of equations:

$$\begin{cases} x - 12 = 4 \\ y - 12 = 36 \end{cases}, \begin{cases} x = 16 \\ y = 48 \end{cases}$$

One of solutions of original equation is $\frac{1}{12} = \frac{1}{16} + \frac{1}{48}$.

E-OLYMP **[6286. Mysterious equation]** Little Vasya is very fond of equations. Once his sight caught the equation $x + y + xy = n$. Vasya wants to know the number of pairs of non-negative integers x and y, that are the solutions of this equation.

► Write the equation in the form

$$(x + 1)(y + 1) = n + 1$$

The number of its solutions equals to the number of divisors of the number $n + 1$. If d is a divisor of $n + 1$, then one of solutions of equation can be found from the system

$$\begin{cases} x + 1 = d \\ y + 1 = (n + 1) / d \end{cases}$$

If we denote by $d(n)$ the number of divisors of the number n, then the answer to the problem will be the value $d(n + 1)$.

Let $n = 5$, consider the equation $x + y + xy = 5$. Rewrite it in the form

$$(x + 1)(y + 1) = 6$$

Number 6 has $d(6) = d(2^1 * 3^1) = 2 * 2 = 4$ divisors (they are 1, 2, 3, 6). To find all solutions to the equation, 4 systems of equations should be solved:

$$\begin{cases} x + 1 = 1 \\ y + 1 = 6 \end{cases}, \begin{cases} x + 1 = 2 \\ y + 1 = 3 \end{cases}, \begin{cases} x + 1 = 3 \\ y + 1 = 2 \end{cases}, \begin{cases} x + 1 = 6 \\ y + 1 = 1 \end{cases}$$

The solutions are the pairs (x, y): (0, 5), (1, 2), (2, 1), (5, 0).

E-OLYMP **[4283. Stepan and pairs]** Given value of n. Find the number of pairs of integers (i, j) such that $1 \le i, j \le n$ and $i = GCD(i, j)$.

► Let us fix j (for example $j = 12$) and find the number of such i that $i = GCD(i, 12)$. Let us find some values of i satisfying this equality. They will be $i = 1, 2, 3, 4, 6, 12$. That is, any i, which is a divisor of 12, satisfies the equality $i = GCD(i, 12)$.

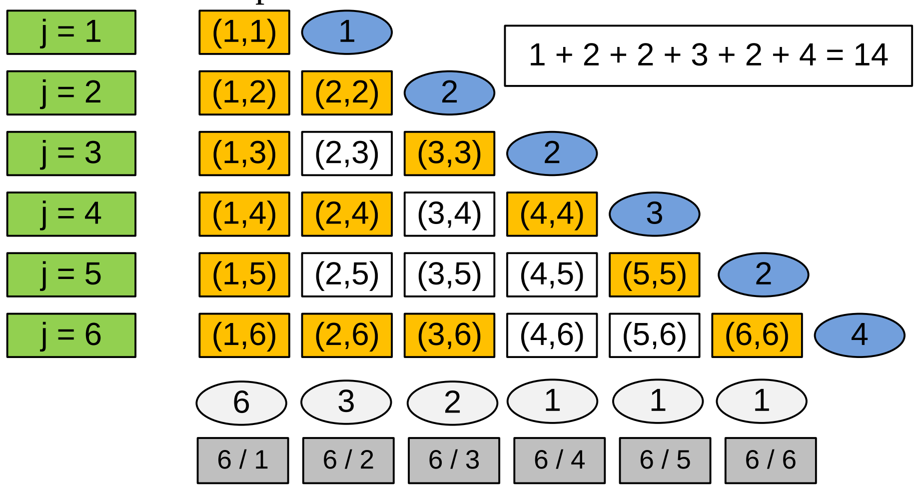
The number of such i for which $i = GCD(i, j)$ equals to the number of divisors $d(j)$ of the number j. Since $1 \le j \le n$, it remains to find the number of all divisors of numbers from 1 to n. That is, you should find the sum $d(1) + d(2) + \ldots + d(n)$. Since the calculation of $d(n)$ requires factorization of the number n, the naive calculation of this sum requires $O(n\sqrt{n})$ time.

Let's look at the sum of divisors in a different way. Among the divisors of numbers from 1 to *n*, one will appear $\lfloor n/1 \rfloor$ times, two will appear $\lfloor n/2 \rfloor$ times, and so on (the divisor *i* will appear $\lfloor n/i \rfloor$ times). The number of required pairs of integers is

$$\sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$$

This sum can be calculated in O(*n*) time.

For *n* = 6 we have the next pairs:



The number of pairs equals to 6 / 1 + 6 / 2 + 6 / 3 + 6 / 4 + 6 / 5 + 6 / 6 =
= 6 + 3 + 2 + 1 + 1 + 1 = 14

E-OLYMP 1782. The sum of divisors Let *n* ($1 \le n \le 10^{18}$) be a positive integer. Find the sum of divisors of number *n*. All its prime divisors do not exceed 1000.

► Let $\sigma(n)$ be the function that computes the sum of all divisors of number *n*.

- If $n = p$ (*p* is prime), then $\sigma(n) = 1 + p$;
- If $n = p^a$ (*p* is prime), then $\sigma(p^a) = 1 + p + p^2 + p^3 + \ldots + p^a = \frac{p^{a+1}-1}{p-1}$;

Function $\sigma(n)$ is multiplicative. If $n = p_1^{a_1} p_2^{a_2} \ldots p_k^{a_k}$, then

$$\sigma(n) = \sigma(p_1^{a_1}) * \sigma(p_2^{a_2}) * \ldots * \sigma(p_k^{a_k}) = \frac{p_1^{a_1+1}-1}{p_1-1} \cdot \frac{p_2^{a_2+1}-1}{p_2-1} \cdot \ldots \cdot \frac{p_k^{a_k+1}-1}{p_k-1}$$

This sum can also be written in the form

$$\sigma(n) = \prod_{i=1}^{k} \frac{p_i^{a_i+1}-1}{p_i-1} = \prod_{i=1}^{k} \left(1 + p_i + p_i^2 + \ldots + p_i^{a_i}\right) = \prod_{i=1}^{k} \sum_{j=0}^{a_i} p_i^j$$

$n = 239$ is prime, $\sigma(239) = 1 + 239 = 240$.
$n = 24 = 2^3 * 3$, $\sigma(24) = \sigma(2^3) * \sigma(3) = (1 + 2 + 4 + 8) * (1 + 3) = 15 * 4 = 60$.
Let's calculate the sum of divisors for $n = 24$ directly: $\sigma(24)$ =
1 + 2 + 3 + 4 + 6 + 8 + 12 + 24 =
1 + 2 + 4 + 8 + 3 * (1 + 2 + 4 + 8) =

$$(1 + 2 + 4 + 8) * (1 + 3) = 15 * 4 = 60$$

Sieve of Eratosthenes is an ancient algorithm for finding all prime numbers up to any given limit. Create a list of consecutive integers from 2 through *n*: (2, 3, 4, ..., *n*). First, numbers greater than 2 and multiples of 2 are removed from the natural series, then numbers greater than 3 and multiples of 3, and so on for each prime number. After the described actions, only prime numbers will remain in the row.

Carry out the described procedure in the array primes[MAX]. First, mark all numbers from 1 to MAX as prime (fill the cells of array primes with 1). Then move through the array from left to right and for each prime number *i*, not greater than $\lfloor\sqrt{MAX}\rfloor$, mark all numbers of the form $i * i$, $i * (i + 1)$, $i * (i + 2)$, … as composite (fill the cells with 0).

As a result of executing the ***gen_primes*** procedure, we get

$$\text{primes}[i] = \begin{cases} 1, \text{if } i \text{ is prime} \\ 0, \text{if } i \text{ is composite} \end{cases}$$

```
void gen_primes(void)
{
  int i, j;
  for(i = 0; i < MAX; i++) primes[i] = 1;
  //primes[0] = primes[1] = 0;
  for(i = 2; i * i < MAX; i++)
    if (primes[i])
      for(j = i * i; j < MAX; j += i) primes[j] = 0;
}
```

Initialize primes array with ones. All integers are declared to be prime.

i	2	3	4	5	6	7	8	9	10	11	12	13	14	15
primes[i]	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Start *for* loop from *i* = 2. primes[2] = 1, so 2 is prime. Start *for j* loop and mark all numbers 2 * 2, 2 * 3, 2 * 4, … as composite.

i	2	3	4	5	6	7	8	9	10	11	12	13	14	15
primes[i]	1	1	0	1	0	1	0	1	0	1	0	1	0	1

Next prime number in the *for* loop is *i* = 3 (primes[3] = 1). Start *for j* loop and mark all numbers 3 * 3, 3 * 4, … as composite.

i	2	3	4	5	6	7	8	9	10	11	12	13	14	15
primes[i]	1	1	0	1	0	1	0	0	0	1	0	1	0	0

In our example MAX = 16, so *for* loop we must iterate till *i* = 4. primes[4] = 0, so 4 is composite. Stop the *for* loop.

The **complexity** of **sieve of Eratosthenes** algorithm is O(n log log n).

E-OLYMP 4739. Sieve of Eratosthenes Given the value of a and b, print all primes in the interval from a to b inclusively.

► Using the Eratosthenes sieve algorithm, fill the primes array, where

- primes[i] = 1, if i is prime;
- primes[i] = 0, if i is composite;

Print all numbers i in the interval from a to b, for which primes[i] = 1.

E-OLYMP 33. The favourite numbers of Santa Claus Santa Claus liked play with numbers and figures. Most of all he liked a number 1 because 1.01 New Year starts.

Years passed, but he stayed be of superstitious – he didn't like numbers, where 3 stand after 1, that is number 13. On New Year he decided to give a new problem: count how many Santa Claus favourite prime numbers contains the interval [a, b]?

► Run the sieve of Eratosthenes for integers up to 500000. For each query, iterate through all numbers from a to b and calculate how many of them are prime and do not contain 13 in decimal notation.

i	...	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
primes[i]	...	0	0	1	0	1	0	0	0	1	0	1	0	0	0	1

In the interval [9; 23] there are 4 prime numbers that do not contain 13 in decimal notation. These numbers are 11, 17, 19, 23.

E-OLYMP 3843. Primes Let m and n ($2 \le m < n \le 10^7$) be two integers. Consider the following set:

$$\text{Prime}(m, n) = \{ p \mid p \text{ prime}, m \le p \le n \}$$

Find the cardinality of the set Prime(m, n).

► Using the sieve of Eratosthenes, fill the array: primes[i] = 1 if i is prime and primes[i] = 0 otherwise. The size of the primes array is 10^7.

Based on the primes array, fill in the cnt array, where cnt[i] contains the number of primes from 1 to i:

- if i is prime, assign cnt[i] = cnt[i – 1] + 1;
- if i is composite, assign cnt[i] = cnt[i – 1];

Then the number of primes in the interval [m; n] equals to cnt[n] – cnt[m – 1].

The filled arrays **primes** and **cnt** have the form:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
primes[i]	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0
cnt[i]	0	1	2	2	3	3	4	4	4	4	5	5	6	6	6

The number of primes in the interval [4; 12] equals to cnt[12] – cnt[3] = 5 – 2 = 3. These primes are 5, 7 and 11.

E-OLYMP **1302. Almost prime numbers** Almost prime numbers are the non-prime numbers which are divisible by only a single prime number. Write a program to find out the number of almost prime numbers within a range [*low*.. *high*] ($0 < low \le high < 10^{12}$).

► *Almost prime* numbers have the form p^k, where p is a prime number, $k \ge 2$. For example, the almost prime numbers are 4, 8, 16, 32, 9, 81, … . As $high < 10^{12}$, then from the definition of almost prime number $p < 10^6$. Generate the array *primes* if length 1000000 = 10^6, where *primes*[*i*] = 1 if *i* is prime and primes[*i*] = 0 otherwise. Then generate and sort in array *m* all the almost primes (their amount equals to 80070).

Let f(*a*, *b*) be the number of *almost primes* in the range [*a*..*b*]. Then

$$f(low, high) = f(1, high) - f(1, low - 1)$$

The number of almost primes in the range [1 .. *n*] equals to the position (index), where it is possible to insert *n* into the sorted array *m* by upper bound preserving the sorted order. The index number can be found using binary search on array *m* by means of function ***upper_bound***.

Let's generate all the almost prime numbers in the range from 1 to 100. First write the powers of 2 not greater than 100. Then the powers of 3, 5 and 7. The square of 11 is greater than 100, so there are no powers of 11 among the almost primes in the range [1..100].

4	8	16	32	64	9	27	81	25	49

Sort the array:

4	8	9	16	25	27	32	49	64	81

Consider the second test case. In the range from 1 to 20 there are 4 almost prime numbers: 4, 8, 9, 16.