

## Основная теорема арифметики. НОД и алгоритм Евклида для его нахождения.

Согласно Основной теореме арифметики любое положительное целое число больше единицы может быть уникально записано в следующей главной форме разложения на множители, где  $p_1, p_2, \dots, p_k$  — простые числа и  $e_1, e_2, \dots, e_k$  — положительные целые числа.

$$n = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_k^{e_k}$$

НОД

$$a = p_1^{a_1} \times p_2^{a_2} \times \dots \times p_k^{a_k}$$

$$b = p_1^{b_1} \times p_2^{b_2} \times \dots \times p_k^{b_k}$$

$$\text{НОД}(a, b) = p_1^{\min(a_1, b_1)} \times p_2^{\min(a_2, b_2)} \times \dots \times p_k^{\min(a_k, b_k)}$$

НОД и НОК связаны друг с другом:  $\text{НОД}(a, b) \times \text{НОК}(a, b) = a \times b$

Наибольший общий делитель двух положительных целых чисел — наибольшее целое число, которое делит оба целых числа.

## Алгоритм Евклида

Нахождение наибольшего общего делителя (НОД) двух положительных целых чисел путем составления списка всех общих делителей непригодно для достаточно больших чисел.

Факт 1:  $\text{НОД}(a, 0) = a$

Факт 2:  $\text{НОД}(a, b) = \text{НОД}(b, r)$ , где  $r$  - остаток от деления  $a$  на  $b$

Реализация алгоритма на языке C++

```

#include <iostream>

int main() {
    int a, b;

    std::cout << "Insert numbers a and b (example: \"24 2\")" << std::endl;
    std::cin >> a >> b;

    std::cout << "GCD of numbers " << a << " and " << b << " = ";

    if (a != b) {
        while (a != 0 && b != 0) {
            if (a > b) {
                a %= b;
            } else {
                b %= a;
            }
        }
        std::cout << a + b;
    } else {
        std::cout << a;
    }

    return 0;
}

```

Операция по модулю - операция, которая показывает остаток от деления одного числа на другое

```

#include <iostream>

int mod(int a, int b) {
    const auto p = a / b;

    const auto mod = a - b * p;

    return mod;
}

int main() {
    std::cout << "Insert 2 numbers to calculate mod" << std::endl;

    int a, b;

    std::cin >> a >> b;

    std::cout << "In-Built operator %: " << a << " % " << b << " = " << a % b
    << std::endl;
    std::cout << "Self wrote mod func: " << a << " % " << b << " = " << mod(a,
    b) << std::endl;

    return 0;
}

```

