1. НОД, простые числа. Решето Эратосфена.

Пусть 𝑎1 , … , 𝑎𝑘 , 𝑑 ∈ ℕ. Если 𝑑 | 𝑎1 , … , 𝑑 | 𝑎𝑘 , то говорят,

что 𝑑 есть общий делитель 𝑎1 , … , 𝑎𝑘 . Наибольший среди всех делителей на-

зывают **наибольшим общим делителем (НОД)** и обозначают 𝐷(𝑎1 , … , 𝑎𝑘 ) или

(𝑎1 , … , 𝑎𝑘 ).

**Наибольший общий делитель (НОД)** двух или более целых чисел — это наибольшее целое число, которое делит каждое из данных чисел без остатка. Например, НОД чисел 8 и 12 равен 4.

**Простое число** — это натуральное число, большее 1, которое имеет ровно два различных натуральных делителя: 1 и само себя. Например, числа 2, 3, 5, 7, 11 являются простыми.

Числа 𝑎1 , … , 𝑎𝑘 называют взаимно простыми, если

𝐷(𝑎1 , … , 𝑎𝑘 ) = 1.

Установим некоторые свойства наибольшего общего делителя и наименьше-

го общего кратного.

Теорема 1.3. Справедливы следующие утверждения:

1) 𝑑 = 𝐷(𝑎1 , … , 𝑎𝑘 ) ⇔ 𝐷 (𝑎1/d, …, 𝑎𝑘/d) = 1;

2) 𝑑 = 𝐷(𝑎1 , … , 𝑎𝑘 ), тогда 𝐷(𝑎1 𝑏, … , 𝑎𝑘 𝑏) = 𝑑𝑏;

3) если 𝑐 – общий делитель 𝑎1 , … , 𝑎𝑘 , то 𝐷 (𝑎1/d, …,𝑎𝑘/𝑑) = d/c;

4) 𝑎𝑏 = 𝐷(𝑎, 𝑏)𝑀 (𝑎, 𝑏).

Справедливо равенство

𝐷 (𝑎1 , 𝑎2 , 𝑎3 ) = 𝐷 (𝐷 (𝑎1 , 𝑎2 ) , 𝑎3 ) .

Определение 1.5. Натуральное число 𝑝 > 1 называют простым, если оно

делится только на ±𝑝 и на ±1.

Теорема 1.7. Пусть 𝑝 – простое число. Справедливо утверждение: 𝑝 | 𝑎𝑏

тогда и только тогда, когда 𝑝 | 𝑎 или 𝑝 | 𝑏.

Лемма 1.1. Любое 𝑎 ∈ ℤ, 𝑎 ≠ 1 имеет по крайней мере один простой дели-

тель.

Теорема 1.8 (oсновная теоремa теории делимости). Любое число

𝑎 ∈ ℤ раскладывается и только одним способом на простые сомножители. Соеди-

нив одинаковые множители в степени, получаем каноническое разложение

𝑎 = 𝑝𝛼 𝑞𝛽𝑟𝛾 ⋯ , где 𝑝, 𝑞, 𝑟 − простые числа; 𝛼, 𝛽, 𝛾 ⩾ 1.

**Решето Эратосфена** — это древний алгоритм для нахождения всех простых чисел до заданного числа n. Алгоритм работает следующим образом:

1. Создаем список последовательных целых чисел от 2 до n.
2. Начинаем с первого числа в списке (2) и удаляем все его кратные (4, 6, 8, ...).
3. Переходим к следующему не удаленному числу и повторяем процесс.
4. Продолжаем до тех пор, пока не пройдем все числа до n.

Найдем все простые числа до 30:

1. Создаем список чисел от 2 до 30.
2. Начинаем с 2, удаляем все кратные 2 (4, 6, 8, ..., 30).
3. Переходим к следующему числу 3, удаляем все кратные 3 (9, 15, 21, 27).
4. Переходим к следующему числу 5, удаляем все кратные 5 (25).
5. Оставшиеся числа (2, 3, 5, 7, 11, 13, 17, 19, 23, 29) являются простыми.

Таким образом, простые числа до 30: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.

**2. Алгоритмы факторизации (метод пробных делителей и метод Ферма).**

Факторизация числа — это процесс разложения числа на произведение простых множителей. Существует несколько методов факторизации, два из которых — **метод пробных делителей** и **метод Ферма** — рассмотрены ниже.

### 1. Метод пробных делителей (Trial Division)

Это самый простой и наивный метод факторизации. Идея заключается в том, чтобы последовательно делить число на все возможные делители, начиная с наименьшего простого числа.

#### Алгоритм:

1. Начинаем с наименьшего простого числа p=2.
2. Делим число n на p:
   * Если n делится на p без остатка, то p является делителем. Записываем p как множитель и делим n на p.
   * Если n не делится на p, увеличиваем p на 1 и повторяем процесс.
3. Процесс продолжается до тех пор, пока p2≤n.
4. Если после завершения цикла n>1, то n само является простым множителем.

#### Пример:

Разложим число 60 на множители:

1. Делим на 2: 60/2=30. Множитель: 2.
2. Делим на 2: 30/2=15. Множитель: 2.
3. Делим на 3: 15/3=5. Множитель: 3.
4. Делим на 5: 5/5=1. Множитель: 5.
5. Результат: 60=2×2×3×5.

### 2. Метод Ферма (Fermat's Factorization Method)

Метод Ферма основан на представлении нечётного числа n в виде разности квадратов:

n=a2−b2

Тогда n можно разложить на множители:

n=(a−b)(a+b)

#### Алгоритм:

1. Находим наименьшее целое a, такое что a2≥n.
2. Вычисляем b2=a2−n.
3. Проверяем, является ли b2 полным квадратом:
   * Если да, то n=(a−b)(a+b).
   * Если нет, увеличиваем a на 1 и повторяем процесс.
4. Процесс продолжается до тех пор, пока не найдётся подходящее b.

#### Пример:

Разложим число n=8051:

1. Находим a: a=⌈√8051⌉=90.

* Вычисляем b2=902−8051=8100−8051=49.
* Проверяем, является ли b2 полным квадратом: 49=72.
* Находим множители: a−b=90−7=83, a+b=90+7=97.
* Результат: 8051=83×97.

#### 3. Алгоритм Евклида.

#### Алгоритм Евклида для нахождения НОД

Алгоритм Евклида — это эффективный способ нахождения НОД двух чисел. Он основан на следующем свойстве: НОД(a, b) = НОД(b, a mod b), где "mod" обозначает операцию взятия остатка от деления.

**Пример:**  
Найдем НОД(48, 18):

1. 48 mod 18 = 12
2. Теперь находим НОД(18, 12):
   * 18 mod 12 = 6
3. Теперь находим НОД(12, 6):
   * 12 mod 6 = 0
4. Когда остаток равен 0, НОД равен последнему ненулевому остатку, то есть 6.

Таким образом, НОД(48, 18) = 6.

**4. Бинарный алгоритм Евклида.**

Бинарный алгоритм Евклида — это оптимизированная версия классического алгоритма Евклида для нахождения наибольшего общего делителя (НОД) двух чисел. Он использует следующие свойства:

1. Если оба числа четные, то НОД также будет четным, и его можно вычислить как:

НОД(a,b)=2⋅НОД(a/2​,b/2​)

1. Если одно из чисел четное, а другое нечетное, то НОД не изменится, если четное число разделить на 2:

НОД(a,b)=НОД(a/2​,b)(если a четное, b нечетное)

1. Если оба числа нечетные, то НОД можно вычислить как:

НОД(a,b)=НОД(∣a−b∣/2​,min(a,b))

#### Преимущества бинарного алгоритма:

* Уменьшает количество операций деления и умножения, заменяя их на битовые сдвиги и проверки на четность.
* Эффективен для больших чисел, особенно на компьютерах, где битовые операции выполняются быстро.

### Алгоритм

1. **Вход:** Два целых неотрицательных числа a и b.
2. **Выход:** НОД(a,b).

#### Шаги:

1. Если a=0, вернуть b.
2. Если b=0, вернуть a.
3. Найти наибольшую степень 2, которая делит и a, и b. Пусть это будет k (можно использовать битовые операции для нахождения k).
4. Разделить a и b на 2k (сдвиг вправо на k бит).
5. Теперь хотя бы одно из чисел a или b нечетное. Пока a≠0:
   * Если a четное, разделить a на 2 (сдвиг вправо на 1 бит).
   * Если b четное, разделить b на 2 (сдвиг вправо на 1 бит).
   * Если оба числа нечетные, заменить большее число на ∣a−b∣/2​.
6. Вернуть b⋅2k как результат.

### Пример

Найдем НОД(48,18) с помощью бинарного алгоритма Евклида:

1. a=48, b=18.
2. Оба числа четные. Находим k=1 (оба делятся на 2).
3. Делим a и b на 2:
   * a=24, b=9.
4. Теперь a четное, b нечетное. Делим a на 2:
   * a=12, b=9.
5. a четное, b нечетное. Делим a на 2:
   * a=6, b=9.
6. a четное, b нечетное. Делим a на 2:
   * a=3, b=9.
7. Оба числа нечетные. Заменяем большее число на ∣3−9∣/2​=3:
   * a=3, b=3.
8. Оба числа равны. Возвращаем b⋅2k=3⋅21=6.

Результат: НОД(48,18)=6.