УО «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра «Информационных систем и технологий»

Специальность 1-40 05 01 «Информационные Системы и Технологии»

**Основы информационной безопасности**

**Практическое задание № 6**

**Вариант 17**

|  |  |  |  |
| --- | --- | --- | --- |
| **Выполнил** |  |  |  |
| Студент 2 курса группы 1 |  |  | Д.И. Велютич |
|  | подпись, дата |  | инициалы и фамилия |
| **Проверил(а)** |  |  | Н.В. Ржеутская |
|  | подпись, дата |  | инициалы и фамилия |

Минск 2023

**Тема «Теория чисел»**

Цель: получение основных сведений из курса теории чисел.

**Теория**

Хорошо, вот пересказ теории о целых числах, как будто ее пересказывает человек:

Целые числа - это числа, которые можно записать без дробной части. Они включают в себя натуральные числа (1, 2, 3, ...), отрицательные числа (-1, -2, -3, ...) и ноль.

На целых числах определены две операции: сложение и умножение. Эти операции обладают следующими свойствами:

* Ассоциативность:
  + (a + b) + c = a + (b + c)
  + (a \* b) \* c = a \* (b \* c)
* Коммутативность:
  + a + b = b + a
  + a \* b = b \* a
* Существует нейтральный элемент:
  + a + 0 = a
  + a \* 1 = a
* Закон дистрибутивности:
  + a \* (b + c) = a \* b + a \* c

Наиболее важным свойством целых чисел является деление с остатком. При делении целого числа a на целое число b остаток r всегда существует и удовлетворяет следующим условиям:

* 0 ≤ r < b
* a = qb + r

Это свойство позволяет нам представлять целые числа в позиционной системе счисления. В этой системе каждое целое число записывается в виде суммы степеней некоторого основания b, где каждый коэффициент - это целое число, не превосходящее b - 1. Например, число 123 в десятичной системе счисления можно представить как 1 \* 10^2 + 2 \* 10^1 + 3 \* 10^0 = 123.

Другим важным свойством целых чисел является наибольший общий делитель. Наибольшим общим делителем двух целых чисел a и b называется такое целое число d, которое делится на a и b, и на которое делится любое другое число, делящееся на a и b. Наибольший общий делитель двух целых чисел можно найти с помощью алгоритма Евклида.

Простое число - это целое число, которое делится только на себя и на единицу. Любое целое число, большее 1, либо является простым, либо имеет простой делитель.

Взаимно простые числа - это два целых числа, у которых наибольший общий делитель равен 1.

Основная теорема арифметики утверждает, что любое целое число, большее 1, можно представить в виде произведения простых чисел.

Сравнение по модулю - это утверждение о том, что два целых числа имеют одинаковые остатки от деления на некоторое целое число m. Сравнения по модулю обладают рядом свойств, которые позволяют нам производить с ними различные операции.

Вот некоторые примеры использования теории целых чисел:

* В математике теория целых чисел используется для решения задач о делении, нахождении наибольшего общего делителя, разложении на простые числа и т.д.
* В информатике теория целых чисел используется для представления и обработки чисел в компьютерах.
* В физике теория целых чисел используется для описания квантовых явлений.

1. Алгоритм Евклида нахождения наибольшего общего делителя (НОД) двух целых чисел a и b работает следующим образом:

- Начните с исходных чисел a и b.

- Пока b не станет равным нулю, выполняйте следующие действия:

- Вычислите остаток r от деления a на b: r = a % b.

- Замените a на b и b на r.

- Когда b станет равным нулю, значение a будет НОД(a, b).

2. Расширенный алгоритм Евклида - это модифицированная версия алгоритма Евклида, которая также находит коэффициенты u и v так, что a \* u + b \* v = НОД(a, b). Этот алгоритм помогает найти соотношение Безу для данных чисел a и b, где НОД(a, b) - это наибольший общий делитель.

3. Два целых числа a и b называются взаимно простыми, если их наибольший общий делитель (НОД) равен 1. Это означает, что у них нет общих положительных делителей, кроме 1.

4. Малая теорема Ферма - это утверждение в теории чисел, сформулированное Леонгардом Эйлером, но названное в честь математика Пьера Ферма. Сама теорема утверждает следующее:

Если p - простое число, и a - целое число, не делящееся на p, то a^(p-1) при делении на p дает остаток 1, то есть a^(p-1) ≡ 1 (mod p).

Эта теорема имеет множество важных приложений в теории чисел и криптографии, включая тесты на простоту и методы шифрования.

**Условия задачи**

1. Найти канонические разложения чисел а и b.
2. Найти НОД  пользуясь:

a) алгоритмом Евклида,

б) разложением чисел на простые множители.

1. С помощью расширенного алгоритма Евклида найти целые u, v, удовлетворяющие соотношению Безу: au + bv = НОД .
2. Найти остаток от деления данного числа на простое.

**У меня 17 вариант, но из-за отсутсвия его наличия – я буду 3им**

|  |  |
| --- | --- |
| 3 | 1-3. *а* = 660422941, *b* = 36481301.  4. Найти остаток от деления  на 17. |

**Исполнительная часть**

Код:  
import sympy

# Задача 1: Найти канонические разложения чисел a и b

a = 660422941

b = 36481301

canonical\_a = [int(p) for p in sympy.primerange(2, a + 1) if a % p == 0]

canonical\_b = [int(p) for p in sympy.primerange(2, b + 1) if b % p == 0]

print("Задача 1: Каноническое разложение a:", canonical\_a)

print("Задача 1: Каноническое разложение b:", canonical\_b)

# Задача 2a: Найти НОД с использованием алгоритма Евклида

def gcd\_euclidean(a, b):

    while b:

        a, b = b, a % b

    return a

gcd = gcd\_euclidean(a, b)

print("Задача 2a: НОД (алгоритм Евклида):", gcd)

# Задача 2б: Найти НОД, разлагая числа на простые множители

def prime\_factors(n):

    factors = []

    i = 2

    while i \* i <= n:

        if n % i:

            i += 1

        else:

            n //= i

            factors.append(i)

    if n > 1:

        factors.append(n)

    return factors

prime\_factors\_a = prime\_factors(a)

prime\_factors\_b = prime\_factors(b)

gcd = 1

for factor in prime\_factors\_a:

    if factor in prime\_factors\_b:

        gcd \*= factor

print("Задача 2б: НОД (разложение на простые множители):", gcd)

# Задача 3: Найти целые u и v, удовлетворяющие соотношению Безу (au + bv = НОД)

def extended\_gcd(a, b):

    if a == 0:

        return (b, 0, 1)

    else:

        g, x, y = extended\_gcd(b % a, a)

        return (g, y - (b // a) \* x, x)

gcd, u, v = extended\_gcd(a, b)

print("Задача 3: НОД:", gcd)

print(f"Задача 3: u: {u}, v: {v}, так что {a} \* {u} + {b} \* {v} = {gcd}")

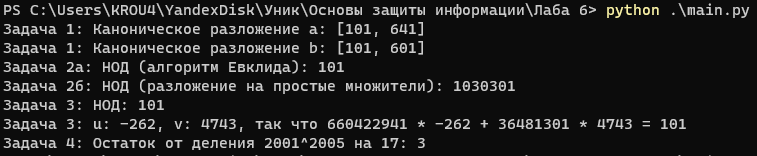
# Задача 4: Найти остаток от деления числа 2001^2005 на 17

result = (2001 \*\* 2005)

result1 = (result % 17)

print("Задача 4: Число 2001^2005:", result)

print("Задача 4: Остаток от деления 2001^2005 на 17:", result1)

Вывод программы:  


**Вывод**

Я успешно достиг цели, получив основные сведения из курса теории чисел. Этот опыт позволил мне углубить свои знания в области математики и развить навыки анализа чисел, применяемые в различных областях науки и инженерии. Полученные знания будут полезными как в моих учебных, так и в профессиональных практических целях.