Практичне завдання

на тему:

**“Теорія чисел. Модульна арифметика”**

**Мета роботи:** опанувати модульну арифметику, алгоритми пошуку найбільшого спільного дільника (НСД) та зворотного елемента, знаходження простих чисел.

# **Теоретичні відомості**

У математиці модульна арифметика являє собою систему розрахунку для цілих чисел, за допомогою якої вони «перевертаються» при досягненні певного значення - модуля. Сучасний підхід до цього виду науки був розвинений Карлом Фрідріхом Гауссом в його книзі Disquisitiones Arithmeticae, опублікованій в 1801 році.

У криптографії закони модульної арифметики, безпосередньо, лежать в основі систем з відкритим ключем, таких як RSA і Діффі-Хелльман. Тут вона надає кінцеві поля, які лежать в основі еліптичних кривих. Використовується в різних алгоритмах симетричного ключа, включаючи Advanced Encryption Standard (AES), Міжнародний алгоритм шифрування даних.

**Піднесення до степеня за модулем**

Приклад: **175235 *mod* 257**

Для виконання такої операції необхідно представити значення степеня як суму степенів двійки або у бінарному вигляді:

23510 = 111010112

*0*. d := 1 \* 175 mod 257 = 175, t := 1752 mod 257 = 42;

*1*. d := 175 \* 42 mod 257 = 154, t := 422 mod 257 = 222;

*2*.(відповідний біт = 0) t := 2222 mod 257 = 197;

*3*. d := 154 \* 197 mod 257 = 12, t := 1972 mod 257 = 2;

*4*. (відповідний біт = 0) t := 22 mod 257 = 4;

*5*. d := 12 \* 4 mod 257 = 48, t := 42 mod 257 = 16;

*6*. d := 48 \* 16 mod 257 = 254, t := 162 mod 257 = 256;

*7*. d := 254 \* 256 mod 257 = 3,

**Алгоритм Евкліда**

Дозволяє знайти НСД двох натуральних чисел. Суть алгоритму Евкліда – два числа порівнюють і від більшого віднімають менше до тих пір, поки числа не стануть рівними. Число, якому вони стануть рівними, і є їх найбільший спільний дільник.

У псевдо-коді такий алгоритм виглядатиме наступним чином:

**функція** нсд(a, b)

**якщо** a = 0

**поверни** b

**поки** b ≠ 0

**якщо** a > b

a := a − b

**інакше**

b := b − a

**поверни** a

**Пошук зворотного елементу**

Для пошуку зворотного елементу необхідно звернутись до розв’язку лінійного рівняння виду: a\*x ≡ 1 (mod n).

Отже, x ≡ *aφ(n)-1* (mod n), якщо *НСД(a,n)=1.*

І для визначення х необхідно розрахувати значення функції Ейлера. Використовуючи властивості функції Ейлера:

* *φ(m\*n) = φ(m)\*φ(n)* якщо *m* i *n* взаємопрості
* *φ(pn) = (pn - pn-1), φ(p)=p-1* якщо *p* просте число

задача зводиться до знаходження всіх простих дільників числа та їх ступенів.

Після чого застосовуючи алгоритм піднесення до степеня за модулем знаходимо розв’язок.

# **Постановка завдання**

Створити програму, що забезпечує виконання арифметичних дій за модулем деякого числа та пошуку зворотнього елемента у відповідній групі(G) .

У програмній реалізації мають бути такі можливості:

1. задавати модуль ***m***, за яким будуть вестись розрахунки
2. розв’язувати рівняння виду **a *mod* m = x**
3. розв’язувати рівняння виду **ab *mod* m = x**
4. розв’язувати рівняння виду **a\**x* ≡ b *mod* m**
5. \* генерувати просте число у діапазоні від **A** до **B**.

# **Вимоги до звіту**

* Для виконання практичного завдання можна використовувати будь-яку зручну для Вас мову програмування.
* Програмний код та особливості щодо запуску/компіляції опублікувати на Github, зробити акаунт **публічним** та надати посилання для перевірки.