**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **КНІ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 1

**З дисципліни:** *“Безпека програм та даних ”*

**На тему:** “Створення генератора псевдовипадкових чисел*”*

**Лектор:**

доцент каф. ПЗ

Сенів М.М.

**Виконав:**

ст. гр. ПЗ-46

Баб’як Д.І.

**Прийняв:**

асист. каф. ПЗ

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2025 р.

∑=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Львів – 2025

**Тема:** Створення генератора псевдовипадкових чисел.

**Мета роботи:** Ознайомитись з джерелами та застосуванням випадкових чисел, алгоритмами генерування псевдовипадкових чисел та навчитись створювати програмні генератори псевдовипадкових чисел для використання в системах захисту інформації.

**Теоретичні відомості**

**1. Вимоги до послідовності випадкових чисел**

**Рівномірність розподілу** – усі числа мають випадати з однаковою ймовірністю.

**Відсутність кореляції** – числа не повинні бути «передбачуваними» одне через одне.

**Великий період** – послідовність не повинна швидко повторюватися.

**Відтворюваність (для ПВЧ)** – при однакових параметрах завжди генерується одна й та сама послідовність (зручно для тестів).

**2. Що таке генератор псевдовипадкових чисел?**

Це алгоритм, який за допомогою математичних формул створює послідовність чисел, що **виглядають випадковими**, хоча вони повністю визначаються початковим значенням (**seed**) і параметрами.

**3. Що таке період послідовності псевдовипадкових чисел?**

Це кількість елементів послідовності до того моменту, коли числа починають **повторюватися по колу**.

**4. Критерії оцінки якості генератора псевдовипадкових чисел**

**Довжина періоду** (чим довший, тим краще).

**Рівномірність розподілу** згенерованих значень.

**Відсутність закономірностей і кореляцій**.

**Швидкість генерації**.

**Простота реалізації** (менше ресурсів, менше коду).

**5. Якими повинні бути параметри ЛКГ для повного періоду**

(для формули Xn+1=(aXn+c) mod mX\_{n+1} = (aX\_n + c) \bmod mXn+1​=(aXn​+c)modm):

ccc і mmm – взаємно прості.

a−1a - 1a−1 кратне всім простим дільникам mmm.

Якщо mmm кратне 4, то й a−1a - 1a−1 теж має бути кратне 4.

**6. Переваги алгоритму лінійного конгруентного (порівняння)**

Дуже простий у реалізації.

Швидко працює навіть для великих обчислень.

Легко відтворити ті самі послідовності (контрольованість).

Використовується у практиці (моделювання, ігри, тести).

**7. Недоліки алгоритму лінійного конгруентного**

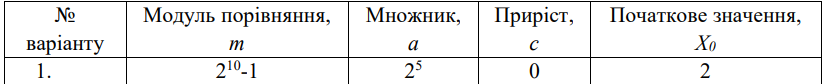
Якщо параметри підібрані неправильно → малий період.

Є статистичні закономірності (послідовність не ідеально «випадкова»).

Не підходить для криптографії (легко відновити наступні числа, знаючи параметри).

**Завдання до виконання роботи**

Згідно до варіанту, наведеного в таблиці, створити програмну реалізацію генератора псевдовипадкових чисел за алгоритмом лінійного порівняння. Програма повинна генерувати послідовність із заданої при вводі кількості псевдовипадкових чисел, результати повинні як виводитись на екран, так і зберігатись у файл. Перевірити період функції генерації, зробити висновок про адекватність вибору параметрів алгоритму. У звіті навести протокол роботи програми, значення періоду функції генерації та зробити висновок про придатність цього генератора для задач криптографії.



**Хід виконання роботи**

**Код програми**

using System;

using System.Collections.Generic;

using System.IO;

using System.Windows;

namespace SPDLab1

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Generate\_Click(object sender, RoutedEventArgs e)

{

try

{

long modulus = long.Parse(ModulusBox.Text);

long multiplier = long.Parse(MultiplierBox.Text);

long increment = long.Parse(IncrementBox.Text);

long seed = long.Parse(SeedBox.Text);

int n = int.Parse(NBox.Text);

var (sequence, period) = GenerateToFile(modulus, multiplier, increment, seed, n);

SequenceList.ItemsSource = sequence; // Для ListBox

PeriodText.Text = $"Період: {period}, Елементів: {sequence.Count}";

}

catch (Exception ex)

{

MessageBox.Show("Помилка: " + ex.Message);

}

}

private (List<long> Sequence, int Period) GenerateToFile(

long modulus, long multiplier, long increment, long seed, int n)

{

string fileName = $"lcg\_{modulus}\_{multiplier}\_{increment}\_{seed}.txt";

string filePath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, fileName);

var sequence = new List<long>();

long x = seed;

int period = 0;

using (var writer = new StreamWriter(filePath, false))

{

do

{

if (sequence.Count < n)

sequence.Add(x);

writer.WriteLine(x);

x = (multiplier \* x + increment) % modulus;

period++;

} while (x != seed && period < modulus);

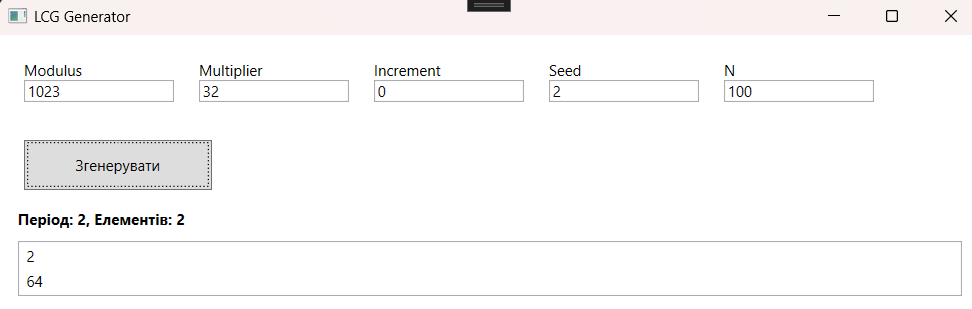
}

return (sequence, period);

}

}

}

****

*Рис. 1. Результат виконання прогарми*

**Висновки**