Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання лабораторної роботи № 1**

**«**Створення генератора псевдовипадкових чисел**»**

**з дисципліни «**Безпека програм та даних**»**

**Лектор**

доцент кафедри ПЗ

Сенів М.М.

**Виконав:**

студент групи ПЗ-45

Ріпей Н.П

**Прийняв:**

асистент кафедри ПЗ

Самбір А.А

«\_\_\_» \_\_\_\_\_\_ 2021р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2021

**Тема роботи:** 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.

**TЕОРЕТИЧНІ ВІДОМОСТІ**

Сучасна інформатика широко використовує випадкові числа в різних програмах – від методу Монте-Карло до криптографії. Ряд алгоритмів захисту мережі, заснованих на засобах криптографії, передбачає використання випадкових чисел. Ці застосування висувають дві вимоги до послідовності випадкових чисел: випадковість і непередбачуваність.

Джерелами дійсно випадкових чисел потенційно можуть бути фізичні генератори шумів, такі як імпульсні детектори іонізуючого випромінювання, газорозрядні лампи, конденсатори з втратами струму тощо. Однак такі пристрої можуть знайти доволі обмежене застосування в додатках для захисту інформації. Тут існують проблеми як з випадковістю, так і з точністю отриманих таким методом чисел, не кажучи вже про проблеми підключення такого роду пристроїв до кожної системи в мережі.

Криптографічні додатки зазвичай використовують алгоритмічні методи генерування випадкових чисел. Відповідні алгоритми є детермінованими і тому породжують послідовності чисел, які не є статистично випадковими. Однак, якщо алгоритм є достатньо хорошим, породжувані ним послідовності чисел витримують багато тестів на випадковість. Такі числа часто називають псевдовипадковими.

Генератор псевдовипадкових чисел – алгоритм, що генерує послідовність чисел, елементи якої незалежні один від одного і підлягають заданому розподілу. Найбільш популярним алгоритмом для генерування псевдовипадкових чисел є алгоритм, запропонований Лемером, що називається методом лінійного порівняння.

**ЗАВДАННЯ**

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.

Варіант №2



**ХІД ВИКОНАННЯ**

Пишу код мовою C#, що реалізує генератор псевдовипадкових чисел за алгоритмом лінійного порівняння:

class PseudoNumberGenerator

    {

        private ulong \_mod;

        private ulong \_multiplier;

        private ulong \_cummulative;

        private ulong \_startingValue;

        private List<ulong> \_sequence;

        private ulong \_period;

        public PseudoNumberGenerator(ulong mod, ulong multiplier, ulong cummulative, ulong startingValue)

        {

            \_mod = mod;

            \_multiplier = multiplier;

            \_cummulative = cummulative;

            \_startingValue = startingValue;

            \_sequence = new List<ulong>();

            \_period = 0;

        }

        public List<ulong> GenerateNumbers(ulong counter)

        {

            for (ulong i = 0; i < counter; ++i)

            {

                ulong n;

                if (\_sequence.Count == 0)

                {

                    n = \_startingValue;

                }

                else

                {

                    n = \_sequence.Last();

                }

                \_sequence.Add(SequenceFormulaGetNext(n));

            }

            CheckPeriod();

            return \_sequence;

        }

        public string GetPeriod()

        {

            return \_period != 0 ? $"Period = {\_period}" :

                "Period hasn't been found, try generating more numbers";

        }

        private ulong SequenceFormulaGetNext(ulong currentNumber)

        {

            return (\_multiplier \* currentNumber + \_cummulative) % \_mod;

        }

        private void CheckPeriod()

        {

            int duplicationIndex = \_sequence.IndexOf(\_sequence.First(), 1);

            \_period = duplicationIndex > 0 ? (ulong)duplicationIndex : 0;

        }

    }

**РЕЗУЛЬТАТИ ВИКОНАННЯ**

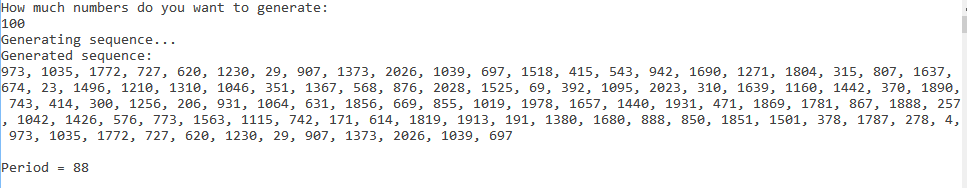


Рис. 1. Результат виконання програми

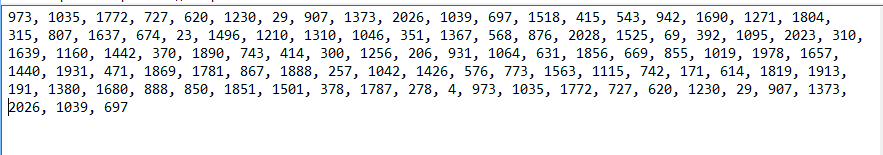


Рис. 2. Файл із послідовністю псевдовипадкових чисел

**ВИСНОВКИ**

На цій лабораторній роботі я ознайомився з джерелами та застосуванням випадкових чисел, дізнався про алгоритми генерування псевдовипадкових чисел. Реалізував алгоритм генерування псевдови-падкових чисел за методом лінійного порівняння. Для параметрів 2 варіанту, значення періоду функції генерації буде 88. Отже, можна дійти до очевидного висновку, що з можливих 2047 чисел використовуються лише 88, це досить малий відсоток унікальних чисел, саме тому ці параметри для генерування псевдовипадкових чисел для практичних задач криптографії не варто застосовувати.