**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **КНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 1

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

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

**Лектор:**

доц. каф. ПЗ

Сенів М. М.

**Виконав:**

ст. гр. ПЗ-44

Вонс Ю. А.

**Прийняв:**

асист. каф. ПЗ

Самбір А. А.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2019 р.

∑= \_\_\_\_\_ .

Львів – 2019

**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**

Генератор псевдовипадкових чисел - алгоритм, що генерує послідовність випадкових чисел, елементи якої незалежні один від одного і підлягають певному розподілу.

Найбільш популярним алгоритмом для генерування псевдовипадкових чисел є алгоритм, запропонований Лемером, який називається методом лінійного порівняння. Алгоритм має чотири наступні параметри:

* m - модуль порівняння, m > 0;
* а - множник, 0 <= a < m;
* c - приріст, 0 <= c < m;
* X0 - початкове число 0 <= X0 < m.

Послідовність псевдовипадкових чисел { X0 } отримують, за допомогою ітерацій наступного співвідношення: Xn+1= (a \* Xn + c) \* mod m

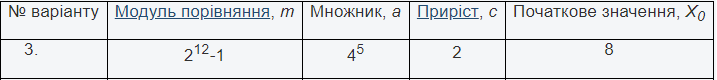
Якщо m, а, c та X0 є цілими, то буде отримано послідовність цілих чисел з діапазону 0 <= Xn < m.

Пропонується три критерії, за якими можна оцінити якість будь-якого генератора псевдовипадкових чисел:

* + - 1. Функція генерації повинна бути функцією повного періоду, тобто функція повинна породити усі числа від 0 до m перед тим, як числа почнуть повторюватись;
      2. Створена послідовність повинна вести себе як випадкова. Насправді ця послідовність не буде випадковою, оскільки генерується детермінованим алгоритмом, але існує багато статистичних тестів, які можна використовувати для того, щоб оцінити ступінь випадковості поведінки послідовності;
      3. Функція генерації повинна ефективно реалізовуватись в рамках 32-бітної арифметики.

**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.

**

**ТЕКСТ ПРОГРАМИ**

1. **Generator.h**

#ifndef GENERATOR\_H

#define GENERATOR\_H

#include <QObject>

#include <cmath>

#include <QString>

class Generator

{

public:

Generator();

void generateValues();

void setSize(uint nValue);

uint getSize();

uint \* getArray();

private:

uint nSize;

uint \*x;

uint m, a , c , X0;

};

#endif // GENERATOR\_H

1. **Generator.cpp**

#include "generator.h"

Generator::Generator()

{

m = pow(2, 12) - 1;

a = pow(4, 5);

c = 2;

X0 = 8;

}

void Generator::generateValues()

{

x = new uint[nSize];

x[0] = X0;

for(uint n = 0; n < nSize - 1; ++n )

x[n+1] = (a\*x[n] + c) % m;

}

void Generator::setSize(uint nValue)

{

nSize = nValue;

}

uint Generator::getSize()

{

return nSize;

}

uint \* Generator::getArray()

{

return x;

}

1. **MainWindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

QFile file("C://Users//hp//Documents//SPD//input.txt");

if (!file.open(QIODevice::ReadOnly | QIODevice::Text))

return ;

QString line;

QTextStream in(&file);

while (!in.atEnd())

line = in.readLine();

ui->lineEdit->setText(line);

obj = new Generator();

}

MainWindow::~MainWindow()

{

delete ui;

delete obj;

}

void MainWindow::on\_pushButton\_clicked()

{

uint size = ui->lineEdit->text().toInt();

if(size > 0) {

obj->setSize(size);

obj->generateValues();

printValues(obj->getArray());

}

}

void MainWindow::printValues(uint \*array)

{

for(uint n = 0; n < obj->getSize(); ++n )

ui->textEdit->insertPlainText(QString::number(array[n]) + ", ");

}

void MainWindow::on\_pushButton\_2\_clicked()

{

ui->textEdit->clear();

ui->lineEdit->clear();

}

void MainWindow::on\_pushButton\_3\_clicked()

{

QFile file("C://Users//hp//Documents//SPD//output.txt");

if (!file.open(QIODevice::WriteOnly | QIODevice::Append))

return;

QTextStream out(&file);

for(uint n = 0; n < obj->getSize(); ++n )

out << QString::number(obj->getArray()[n]) << ", ";

file.close();

}

**РЕЗУЛЬТАТИ**

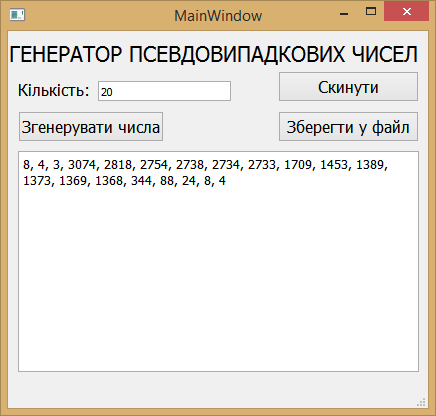


Рис. 1. Результат виконання програми

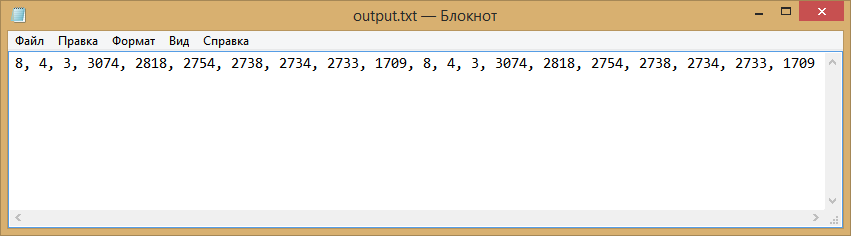


Рис. 2. Результат виконання програми

**ВИСНОВКИ**

Виконуючи дану лабораторну роботу, я ознайомився з джерелами та застосуваннями випадкових чисел та навчився створювати алгоритм для генерування псевдовипадкових чисел, запропонований Лемером, який називається методом лінійного порівняння.

Розроблена програма дозволяє генерувати послідовність із заданої при вводі кількості псевдовипадкових чисел за допомогою алгоритму Лемера та зберігати результати у файл. Період функції генерації за даною умовою рівний 18. Даний генератор псевдовипадкових чисел не є придатним для завдань криптографічний додатків оскільки має надто малий період.