**[Лабораторна робота № 1](http://vns.lp.edu.ua/mod/page/view.php?id=197091" \o "Лабораторна робота № 1)**

**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**

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. Туту 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.

Тому 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](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187762&displayformat=dictionary) – 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, який називається методом лінійного порівняння. Цей алгоритм має чотири наступних параметри.

|  |  |  |
| --- | --- | --- |
| *m* | [модуль порівняння](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187793&displayformat=dictionary) | *m* > 0 |
| *a* | множник | 0 ≤ *a* < *m* |
| *c* | [приріст](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187811&displayformat=dictionary) | 0 ≤ *c* < *m* |
| *X0* | початкове число | 0 ≤ *X0* < *m* |

Послідовність псевдовипадкових чисел {*X0*} отримують за допомогою ітерацій наступного співвідношення:

.

При цьому якщо *m*, *a*, *c* та *X0* є цілими, то буде отримано послідовність цілих чисел з діапазону 0 ≤ *Xn* < *m.*

Вибір значень для *a*, *c* та *m* є дуже важливим з точки зору створення хорошого генератора псевдовипадкових чисел. Розглянемо, наприклад, випадок *a*= *c*= 1. Породжена при цьому послідовність очевидно не буде задовільною. Тепер розглянемо значення *a*= 7, *c*= 0, *m*= 32 *X0*= 1. В цьому випадку генерується послідовність {7, 17, 23, 1, 7, …}, яка також очевидно не буде задовільною. З 32 можливих значень тут виявляються задіяними тільки 4 (в такому випадку кажуть, що послідовність має період 4). Якщо ж, залишивши інші значення тими самими, змінити значення *a* і прийняти *a*=5, то результуючою послідовністю буде {1, 5, 25, 29, 17, 21, 9, 13, 1, …} і її період вже буде рівним 8.

Бажано, щоб *m* було дуже великим, щоб потенційно могли генеруватись дуже довгі серії різних псевдовипадкових чисел. Загальним правилом тут є вибір значення *m*, близького до максимально допустимого для даного комп'ютера додатного цілого числа. Тому доволі часто значення *m* вибирається рівним або майже рівним значенню 231.

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.

Усі ці три критерії можуть задовольнятись при адекватному виборі значень *a*, *c* та *m*. Відносно першого критерію можна довести, що якщо *m* є простим і *c*= 0, то для певних значень *a* період генерованої функцією послідовності виявляється рівним *m*-1 і в цій послідовності буде відсутнім тільки значення 0. В 32-бітній арифметиці зручним простим значенням для *m* є значення 231-1. В цьому випадку функція генерації приймає вигляд

.

З більш ніж двох мільйонів можливих значень *a* тільки декілька множників відповідають функції, що витримує усі три тести. Одним з таких значень є *a* = 75 = 16807, яке було знайдено і використано для родини комп'ютерів IBM 360. Відповідний генератор знайшов широке застосування, і тому він був підданий більш ретельному аналізу, ніж будь-який інший [генератор псевдовипадкових чисел](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187762&displayformat=dictionary). Він нерідко рекомендується для статистичного та імітаційного моделювання різноманітних процесів.

Перевагою алгоритму лінійного порівняння є те, що якщо вибрати адекватні множник та [модуль порівняння](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187793&displayformat=dictionary), то створювана послідовність чисел виявляється статистично невідрізнимою від послідовності чисел, що вибираються випадково (але незворотно) з множини чисел 1, 2, …, *m*-1. Однак в самому алгоритмі немає нічого випадкового взагалі, крім вибору початкового значення *X0*. Якщо це значення вибране, інші числа послідовності визначаються ним однозначно. Це виявляється дуже важливим з точки зору [криптоаналіз](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187777&displayformat=dictionary" \o "Словник основних термінів: Криптоаналіз)у.

Якщо противник знає, що використовується алгоритм лінійного порівняння і якщо до того ж йому відомі параметри алгоритму (наприклад, *a* = 75, *c* = 0, *m* = 231-1), то, відкривши усього одно число, противник може отримати всі наступні. Але якщо навіть опонент знає тільки те, що вибрано алгоритм лінійного порівняння, знання невеликої частини послідовності вже достатньо для того, щоб визначити усі параметри алгоритму. Припустимо, наприклад, що противник зможе визначити значення для *X0*, *X1*, *X2* та *X3*. Тоді

,

,

.

Ці рівняння можуть бути розв'язані відносно *a*, *c* та *m*.

Отже, хоча і зручно використовувати хороший [генератор псевдовипадкових чисел](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187762&displayformat=dictionary), бажано подбати про те, що генерована послідовність була дійсно невідтворюваною, щоб знання частини послідовності не давало опоненту можливості визначити наступні елементи послідовності. Ця мета може бути досягнута цілим рядом способів. Наприклад можна змінювати потік псевдовипадкових чисел, використовуючи для цього системний час. Один зі способів на основі системного годинника полягає в ініціалізації нової послідовності після отримання кожних *N* чисел, використовуючи для початкового числа поточне значення часу (mod *m*). А можна просто додавати до кожного псевдовипадкового числа поточне значення часу (mod *m*).

**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.

Варіанти для виконання лабораторної роботи:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № варіанту | [Модуль порівняння](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187793&displayformat=dictionary), *m* | Множник, *a* | [Приріст](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187811&displayformat=dictionary), *c* | Початкове значення, *X0* |
| 1. | 210-1 | 25 | 0 | 2 |
| 2. | 211-1 | 35 | 1 | 4 |
| 3. | 212-1 | 45 | 2 | 8 |
| 4. | 213-1 | 55 | 3 | 16 |
| 5. | 214-1 | 65 | 5 | 32 |
| 6. | 215-1 | 23 | 8 | 64 |
| 7. | 216-1 | 33 | 13 | 128 |
| 8. | 217-1 | 43 | 21 | 256 |
| 9. | 218-1 | 53 | 34 | 512 |
| 10. | 219-1 | 63 | 55 | 1024 |
| 11. | 220-1 | 73 | 89 | 1 |
| 12. | 221-1 | 83 | 144 | 3 |
| 13. | 222-1 | 93 | 233 | 5 |
| 14. | 223-1 | 103 | 377 | 7 |
| 15. | 224-1 | 113 | 610 | 9 |
| 16. | 225-1 | 123 | 987 | 11 |
| 17. | 226-1 | 133 | 1597 | 13 |
| 18. | 227-1 | 143 | 2584 | 17 |
| 19. | 228-1 | 153 | 4181 | 19 |
| 20. | 229-1 | 163 | 6765 | 23 |
| 21. | 230-1 | 173 | 10946 | 29 |
| 22. | 231-1 | 75 | 17711 | 31 |
| 23. | 231 | 216 | 28657 | 33 |
| 24. | 231-3 | 215 | 46368 | 37 |
| 25. | 231-7 | 214 | 75025 | 41 |

**Контрольні запитання**

1.     Назвіть вимоги до послідовності випадкових чисел.

2.     Що таке [генератор псевдовипадкових чисел](http://vns.lp.edu.ua/mod/glossary/showentry.php?courseid=8932&eid=187762&displayformat=dictionary)?

3.     Що таке період послідовності псевдовипадкових чисел?

4.     Назвіть критерії оцінки якості генератора псевдовипадкових чисел.

5.     Якими повинні бути параметри алгоритму лінійного порівняння для того, щоб генерувався повний період псевдовипадкових чисел?

6.     Назвіть переваги алгоритму лінійного порівняння.

1. Назвіть недоліки алгоритму лінійного порівняння.