Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

**ТЕОРИЯ ПСЕВДОСЛУЧАЙНЫХ ГЕНЕРАТОРОВ**

**ОТЧЕТ ПО ПРАКТИЧЕСКОМУ КУРСУ**

студентки 4 курса 431 группы

факультета компьютерных наук и информационных технологий

*Стыценковой Валерии Сергеевны*

фамилия, имя, отчество

Научный руководитель

Ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_ И.И. Слеповичев

подпись, дата

Саратов 2023

## Задание 1. Генерация псевдослучайных чисел.

**Описание задания:**

Создать программу для генерации псевдослучайных величин числа из заданного диапазона следующими алгоритмами:

* 1. Линейный конгруэнтный метод;
  2. Аддитивный метод;
  3. Пятипараметрический метод;
  4. Регистр сдвига с обратной связью (РСЛОС);
  5. Нелинейная комбинация РСЛОС;
  6. Вихрь Мерсенна;
  7. RC4;
  8. ГПСЧ на основе RSA;
  9. Алгоритм Блюма-Блюма-Шуба;

Название программы: **prng.exe**

Входные параметры алгоритма передаются программе через строку параметров.

Для управления приложением предлагается следующий формат параметров командной строки:

**/g:<код\_метода>** - параметр указывает на метод генерации ПСЧ, при этом код\_метода может быть одним из следующих:

* lc – линейный конгруэнтный метод;
* add – аддитивный метод;
* 5p – пятипараметрический метод;
* lfsr – регистр сдвига с обратной связью (РСЛОС);
* nfsr – нелинейная комбинация РСЛОС;
* mt – вихрь Мерсенна;
* rc4 – RC4;
* rsa – ГПСЧ на основе RSA;
* bbs – алгоритм Блюма-Блюма-Шуба;

Элементы вектора параметров разделяются запятой. Порядок элементов вектора для каждого из методов описан в таблице 1.

Таблица 1. Порядок элементов вектора параметров

|  |  |
| --- | --- |
| **Метод** | **Описание параметров** |
| lc | Модуль, множитель, приращение, начальное значение |
| add | Модуль, младший индекс, старший индекс, последовательность начальных значений |
| 5p | p, q1, q2, q3, w (см. лекции п. 3.5.2), начальное значение |
| lfsr | Двоичное представление вектора коэффициентов, начальное значение регистра |
| nfsr | В алгоритме использовать три РСЛОС R1, R2, R3, скомбинированных функцией R1^R2+R2^R3+R3. Параметры – двоичное представление векторов коэффициентов для R1, R2, R3, w, x1, x2, x3. w – длина слова, x1, x2, x3 – десятичное представление начальных состояний регистров R1, R2, R3. |
| mt | Модуль, начальное значение x |
| rc4 | 256 начальных значений |
| rsa | Модуль n, число e, w, начальное значение x.  e удовлетворяет условиям: 1 < e < (p-1)(q-1), НОД(e, (p-1)(q-1)) = 1, где p\*q=n.  x из интервала [1,n]  w – длина слова. |
| bbs | Начальное значение x (взаимно простое с n).  При генерации использовать параметры:  p = 127, q = 131, n = p\*q = 16637 |

**/n:<длина>** - количество генерируемых чисел. Если параметр не указан, - генерируется 10000 чисел.

**/f:<полное\_имя\_файла>** - полное имя файла, в который будут выводиться данные. Если параметр не указан, данные должны записываться в файл с именем rnd.dat.

**/h** – информация о допустимых параметрах командной строки программы.

Выходные значения записываются в файл .dat, указанный в параметре запуска программы.

### Алгоритм 1. Линейный конгруэнтный метод.

**Описание алгоритма.**

Одним из простых и популярных методов сейчас является *линейный конгруэнтный метод* (ЛКМ), предложенный Д.Г. Лехмером в 1949 году. В его основе лежит выбор четырех ключевых чисел:

* , модуль;
* , множитель;
* , приращение (инкремент);
* , начальное значение.

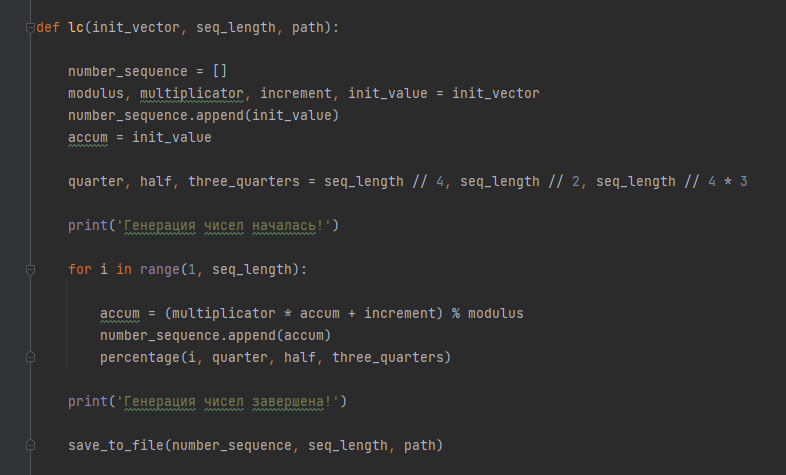
**Определение 3.1.** Последовательность ПСЧ, получаемая по формуле:

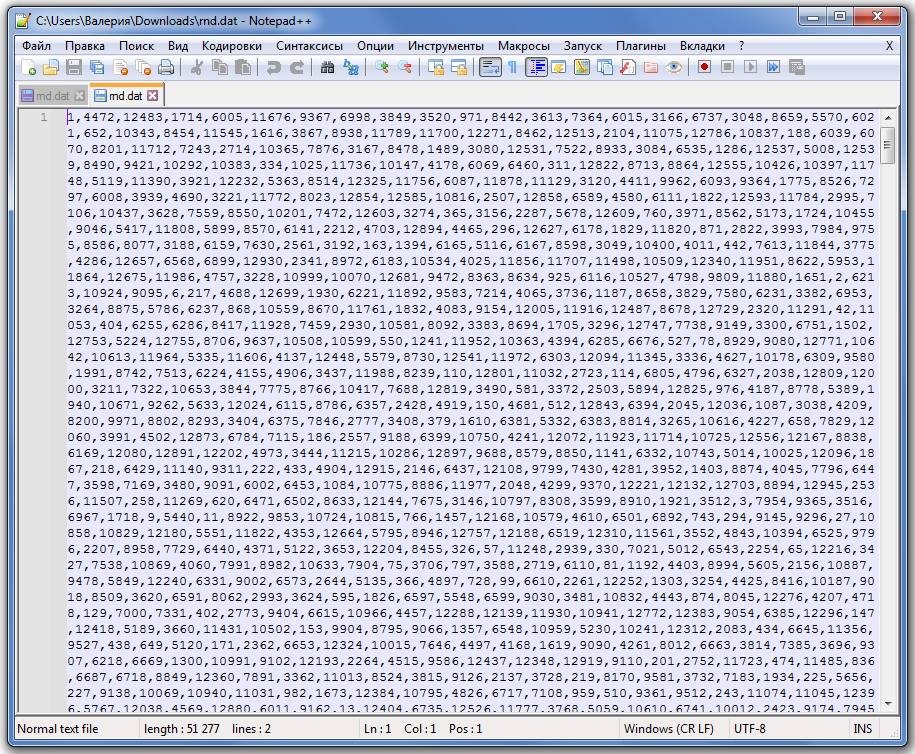
, (3.1)

называется *линейной конгруэнтной последовательностью* (ЛКП). Ключом для неё служит .

**Параметры запуска программы.**

/g:lc /i:12960,1741,2731,1

**Исходный текст программы.  
**

**Пример работы программы.  
**

### Алгоритм 2. Аддитивный метод.

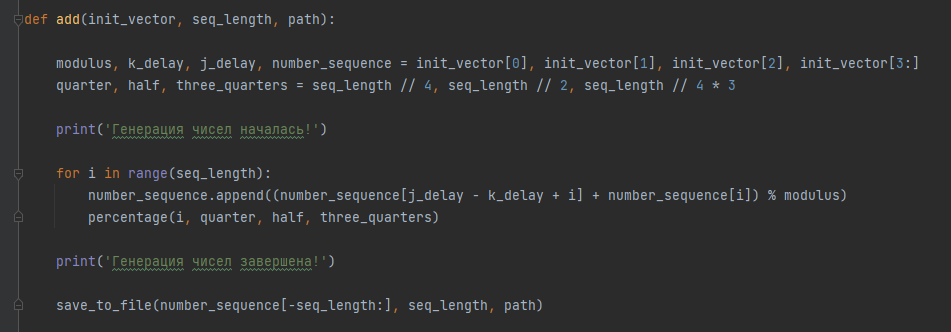
**Описание алгоритма.**

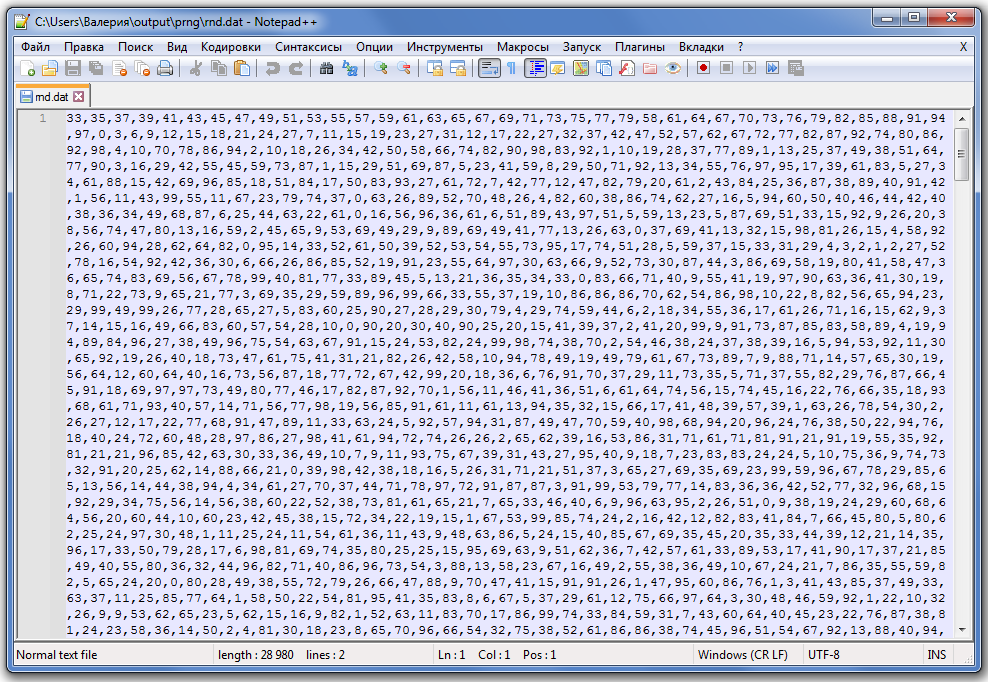
Каждое следующее значение вычисляется по рекуррентной формуле:

Числа – целые числа, которые называются запаздываниями, – это модуль, – длина вектора, который подается на вход, берется из вектора начальных значений.

**Параметры запуска программы.**

/g:add  
/i:100,24,55,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55

**Исходный текст программы.**

**Пример работы программы.  
**

### Алгоритм 3. Пятипараметрический метод.

**Описание алгоритма.**

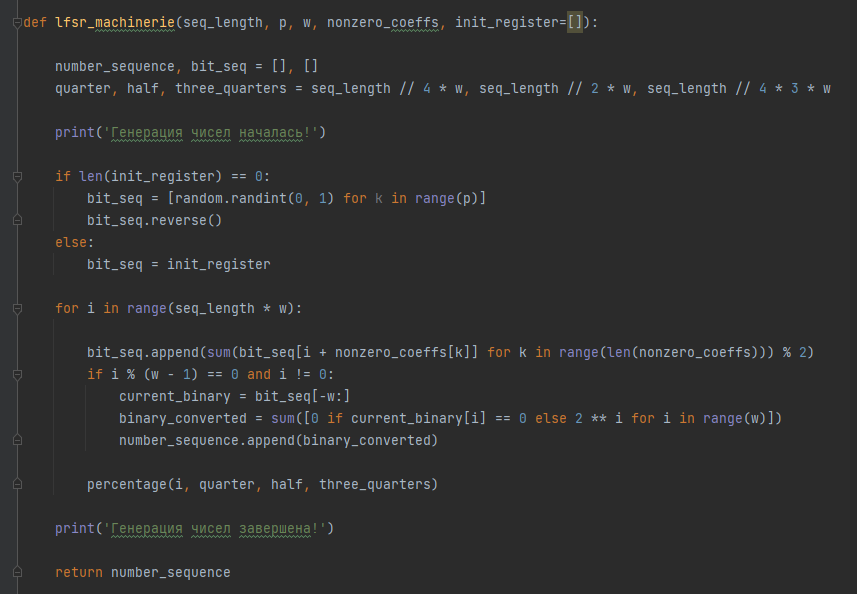
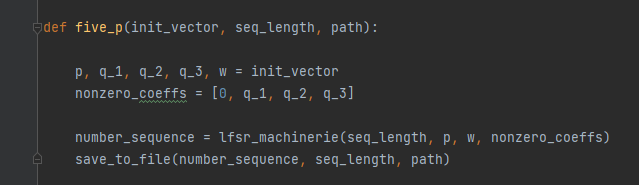
Данный метод является частным случаем РСЛОС, использует характеристический многочлен из 5 членов и позволяет генерировать последовательности *w*-битовых двоичных целых чисел в соответствии со следующей рекуррентной формулой:

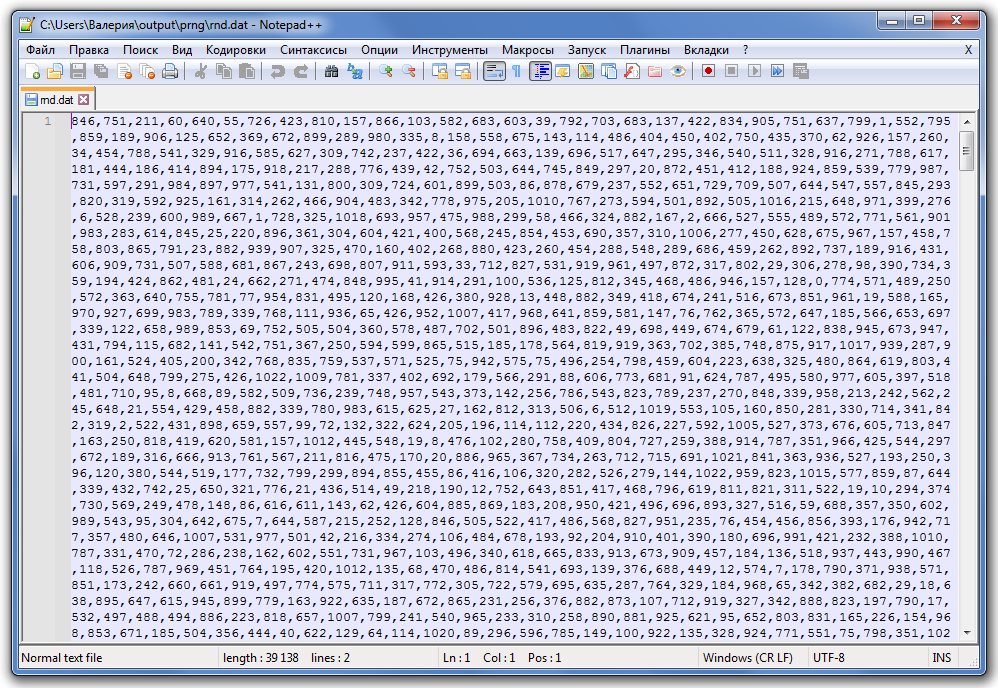
* Параметры и , первоначально задают как начальный вектор.

**Параметры запуска программы.**

/g:5p /i:89,20,40,69,10

**Исходный текст программы.**



**Пример работы программы.  
**

### Алгоритм 4. Регистр сдвига с обратной связью (РСЛОС).

**Описание алгоритма.**

Следующий класс ГПСЧ основан на идее преобразования двоичного представления некоторого числа. Такие генераторы имеют некоторые преимущества, как, например, скорость генерации таких чисел, хорошие статистические свойства ПСЧ, а также возможность простой реализации на аппаратном уровне.

Для натурального числа *p* и , принимающих значения 0 или 1, определяют рекуррентную формулу

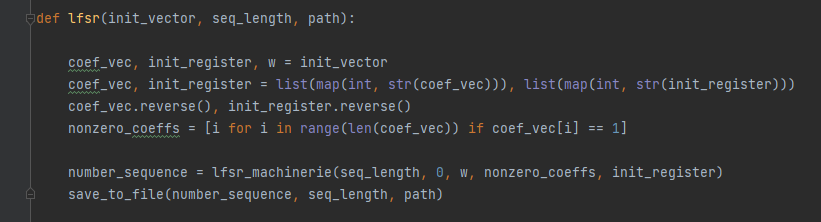
(3.7)

Как видно из формулы, для РСЛОС функция обратной связи является линейной булевой функцией от состояний всех или некоторых битов регистра.

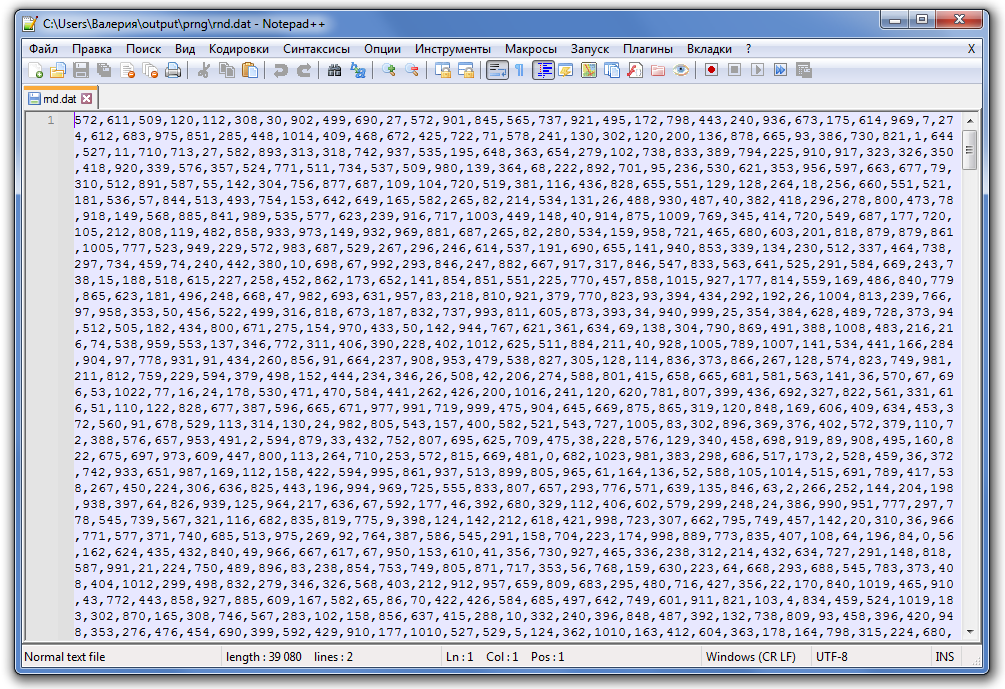
**Параметры запуска программы.**

/g:lfsr /i:1000010001,1010101110100011100010101010,10

**Исходный текст программы.**

****

**Пример работы программы.**



### Алгоритм 5. Нелинейная комбинация РСЛОС.

**Описание алгоритма.**

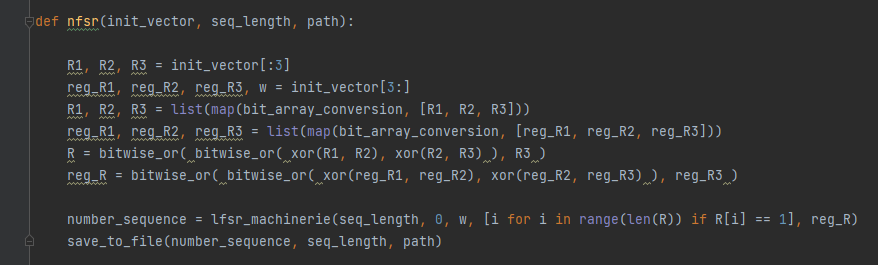
Предположим, что у нас регистров сдвига с линейной обратной связью , их длины попарно различны и больше двух. Подобно комбинированному методу Таусворта, мы можем объединить эти РСЛОС, но уже при помощи нелинейной функции *f:*

.

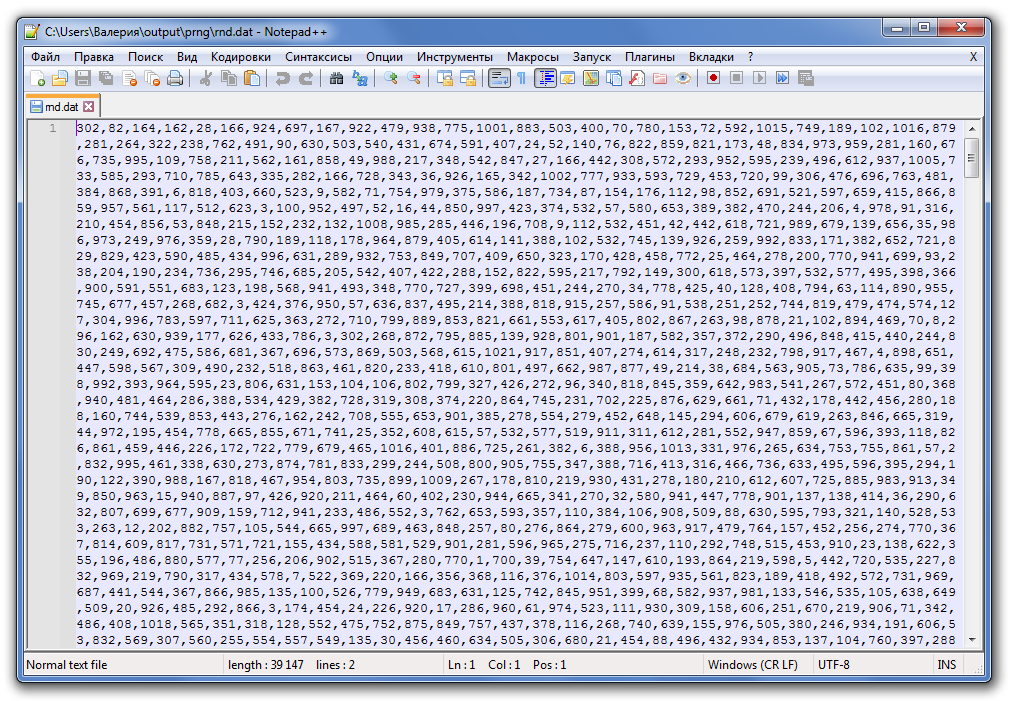
**Параметры запуска программы.**

/g:nfsr  
/i:10101,1000011001,10101000000001,10101001,10101000111001,1010010101010000101011100001,10

**Исходный текст программы.**

****

**Пример работы программы.**



### Алгоритм 6. Вихрь Мерсенна.

**Описание алгоритма.**

Метод Вихрь Мерсенна позволяет генерировать последовательность двоичных псевдослучайных целых w-битных чисел в соответствии с рекуррентной формулой:

где – целые константы;

– степень рекуррентности, ;

– *w*-битное двоичное целое число;

– двоичное целое число, полученное конкатенацией чисел и когда первые битов взяты из , а последние битов из в том же порядке;

– матрица размера состоящая из нулей и единиц, определенная

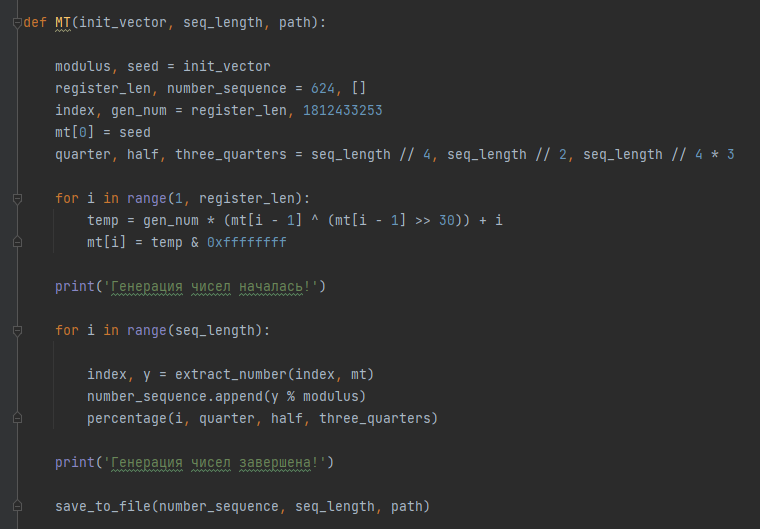
посредством .

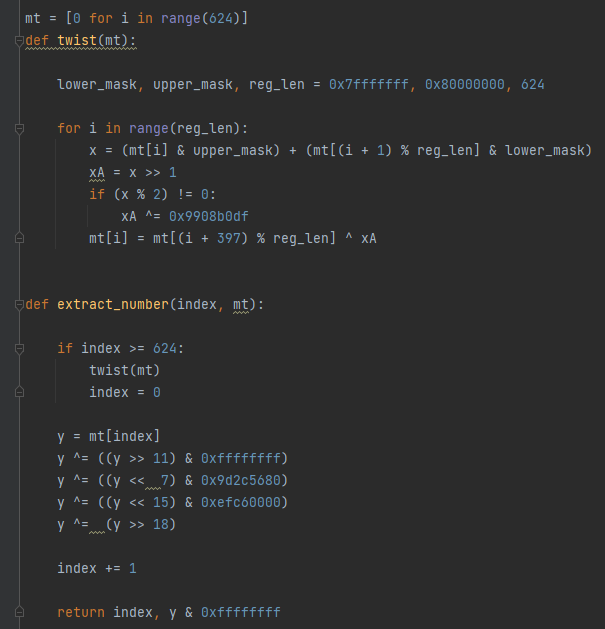
– произведение, при вычислении которого сначала выполняют операцию (сдвига битов на одну позицию вправо), если последний бит равен 0, а затем, когда последний бит , то вычисляют .

**Параметры запуска программы.**

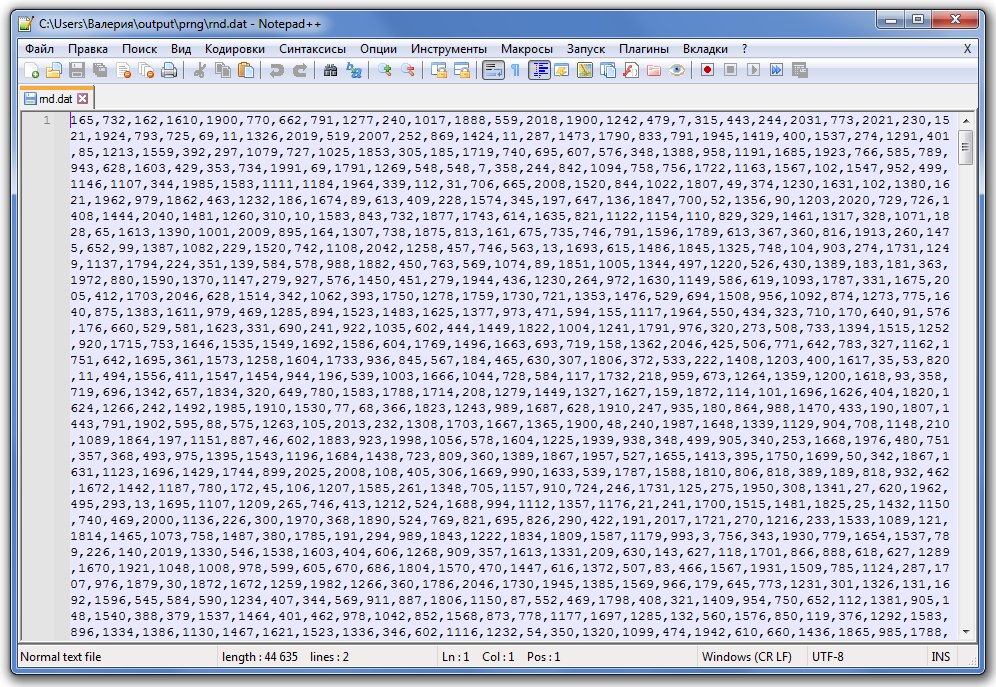
/g:mt /i:2048,113

**Исходный текст программы.**

****

****

**Пример работы программы.**

****

### Алгоритм 7. RC4.

**Описание алгоритма.**

Являясь потоковым шифром, в основе которого генератор псевдослучайных чисел, RC4 широко используется в различных криптографических протоколах. Достоинством алгоритма является высокая скорость работы и переменный размер ключа.

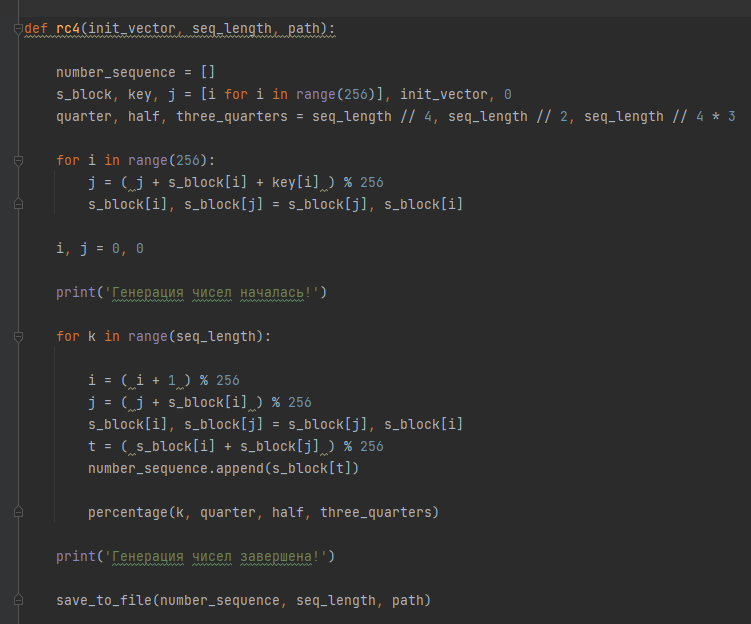
1. Инициализация ,
2. Итерация алгоритма:

Стойкость алгоритма основана на следующем наблюдении: даже если оппонент узнал ключ *К* и номер шага *i*, он может вычислить всего лишь значение , но не всё внутреннее состояние массива. Это следует из того, что оппонент не в состоянии определить значение переменной *t*, не зная *j*, .

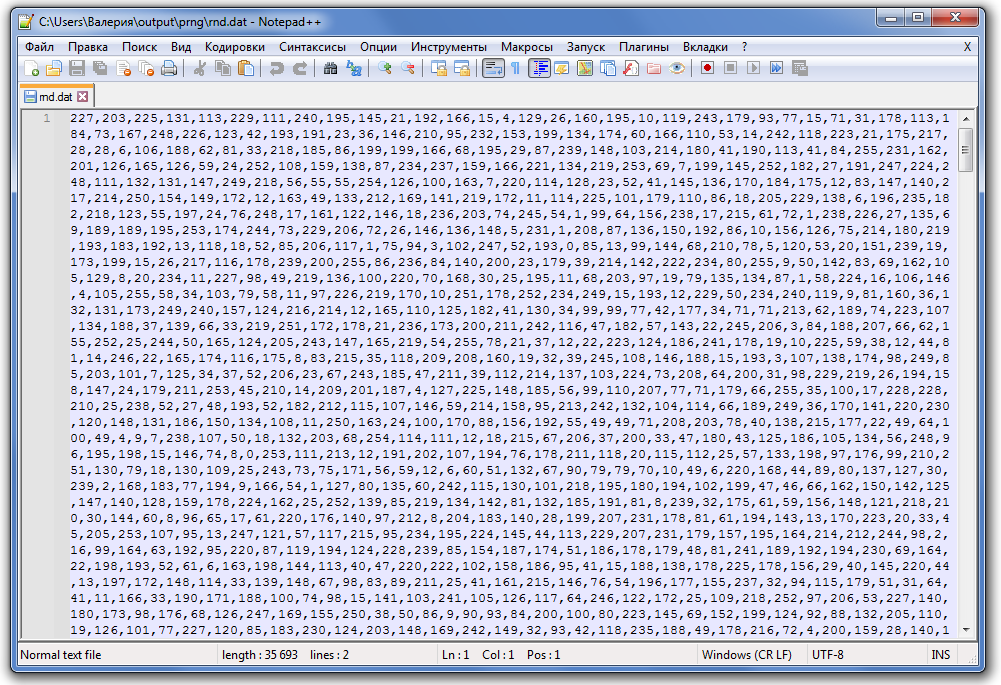
**Параметры запуска программы.**

/g:rc4 /i:281,27,38,169,475,248,398,1,213,431,570,782,743,565,560,427,36,679,321,947,45,141,537,297,498,812,510,824,206,232,266,789,712,473,750,699,731,965,1010,53,97,400,754,597,56,330,462,723,116,30,646,621,64,982,64,77,767,311,464,256,599,647,945,886,825,823,1004,508,576,266,704,977,767,503,116,674,105,994,904,1013,399,535,606,414,1015,165,951,958,52,769,17,1014,952,987,481,772,389,435,1022,998,171,36,159,579,218,174,102,715,898,834,790,888,34,187,1001,108,474,856,365,60,983,144,437,155,270,674,905,334,111,945,380,775,917,792,995,829,855,848,274,645,156,539,11,556,895,980,658,62,81,244,592,89,948,1017,787,749,202,413,840,200,223,92,385,59,725,723,113,1010,490,635,677,729,178,472,124,448,776,624,314,534,379,622,508,471,509,504,428,385,482,174,590,134,920,437,816,301,31,682,471,692,83,342,46,938,647,20,542,647,609,850,888,394,71,700,691,298,1016,146,675,50,701,567,740,692,946,161,675,781,533,485,146,670,236,736,72,626,837,986,935,786,164,922,798,275,601,43,965,737,910,743,117,280,562,944,618,928

**Исходный текст программы.**

****

**Пример работы программы.**

****

### Алгоритм 8. ГПСЧ на основе RSA.

**Описание алгоритма.**

Эди Шамир предложил использовать алгоритм шифрования с открытым ключом RSA для генерации ПСЧ [19]. В его работе показано, что предсказание выхода генератора псевдослучайных чисел равносильно взлому RSA. Очевидным недостатком такого алгоритма является низкая скорость и громоздкость реализации.

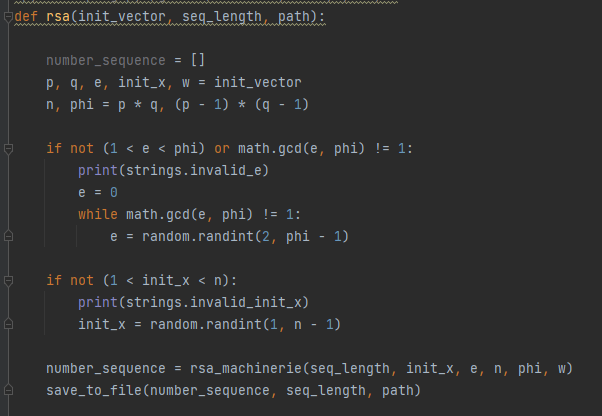
Дальнейшей модификацией этого алгоритма является ГПСЧ RSA, который также основан на сложности решения проблемы RSA.

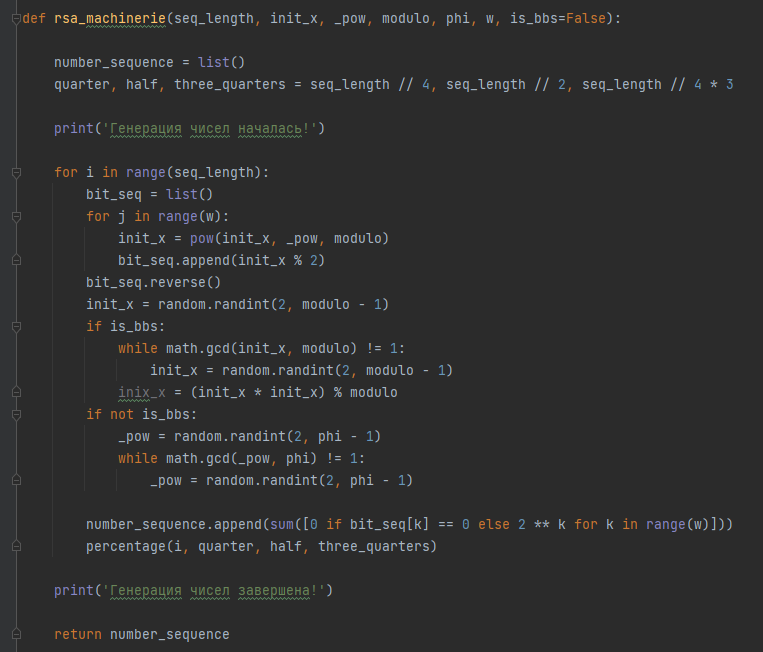
1. Сгенерировать два секретных простых числа *p* и *q*, а также и . Выбрать случайное целое число , такое что
2. Выбрать случайное целое – начальный вектор из интервала
3. .

**Параметры запуска программы.**

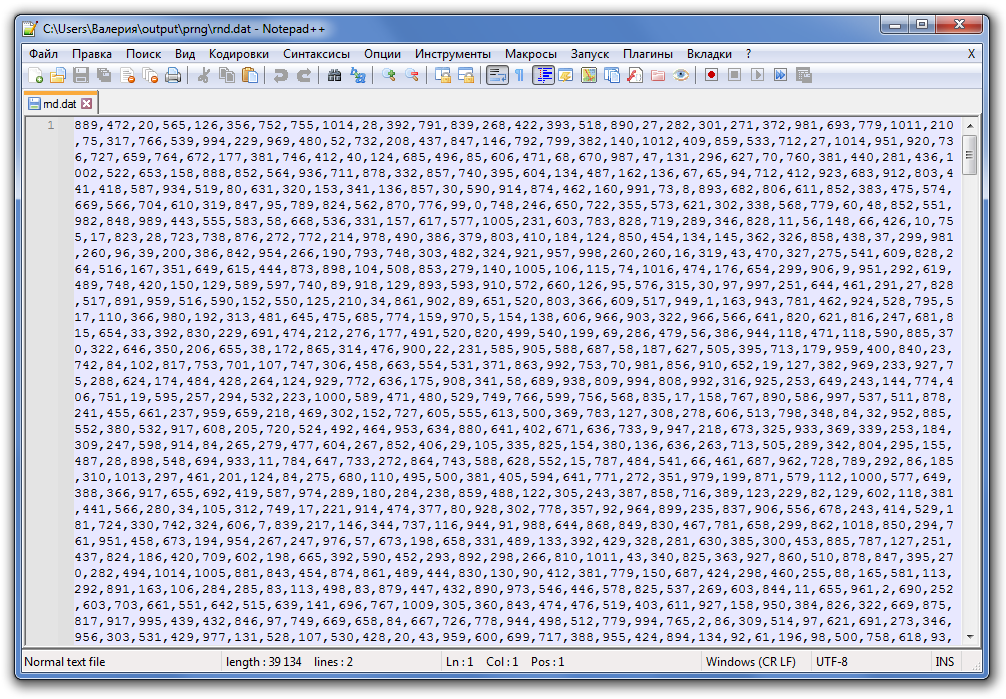
/g:rsa /i:514081,99991,0,127,10

**Исходный текст программы.**

****

****

**Пример работы программы.**

****

### Алгоритм 9. Алгоритм Блюма-Блюма-Шуба.

**Описание алгоритма.**

Алгоритм был предложен в 1986 году. В основе алгоритма – использование квадратичных остатков по модулю *n*. На текущее время это один из самых простых и быстрых алгоритмов ГПСЧ, использующих вычислительно сложные задачи.

**На входе:** Длина *l*.

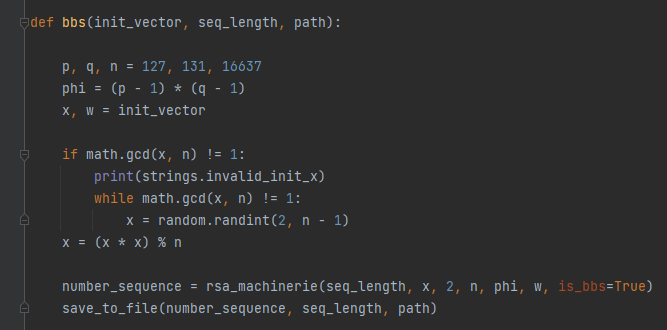
**На выходе:** Последовательность псевдослучайных бит .

1. Сгенерировать два простых числа *p* и *q*, сравнимых с 3 по модулю 4. Это гарантирует, что каждый квадратичный вычет имеет один квадратный корень, который также является квадратичным вычетом. Произведение этих чисел – *n=pq* является целым числом Блюма. Выберем другое случайное целое число *x*, взаимно простое с *n*.
2. Вычислим , которое будет начальным вектором.
3. .

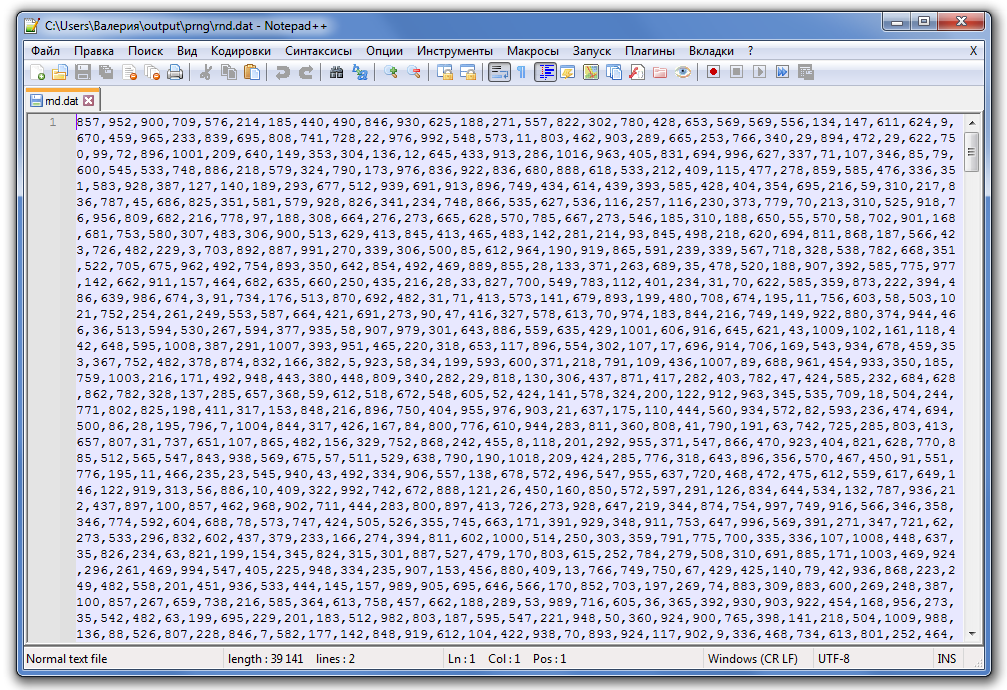
**Параметры запуска программы.**

/g:bbs /i:113,10

**Исходный текст программы.**

****

**Пример работы программы.**

****