**Лабораторная работа №3.** [*Поточные криптосистемы*](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80)

*Введение.* Для выработки ключей криптосистем и гаммы нужно использовать [криптографические генераторы](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8_%D1%81%D1%82%D0%BE%D0%B9%D0%BA%D0%B8%D0%B9_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB) псевдослучайных чисел. Необходимость отказа от обычных программных генераторов (например, от [линейного конгруэнтного метода](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BD%D0%B3%D1%80%D1%83%D1%8D%D0%BD%D1%82%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4)) обусловлена двумя причинами: незащищенностью реализации и возможностью предугадать значение. Выделяют класс криптографических генераторов, базовым элементом которых является [регистр сдвига с линейной обратной связью](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80_%D1%81%D0%B4%D0%B2%D0%B8%D0%B3%D0%B0_%D1%81_%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%BE%D0%B9_%D0%BE%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B9_%D1%81%D0%B2%D1%8F%D0%B7%D1%8C%D1%8E) (РСЛОС). При этом, как правило, требуется, чтобы РСЛОС генерировал [линейную рекуррентную последовательность](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%80%D0%B5%D0%BA%D1%83%D1%80%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C) (ЛРП) с максимально возможным периодом (относительно *длины* *РСЛОС* (количества ячеек памяти у РСЛОС)), высокой линейной сложностью и хорошими статистическими свойствами. Так как ЛРП не обладает нелинейными свойствами, то возможно [восстановление РСЛОС](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%91%D0%B5%D1%80%D0%BB%D0%B5%D0%BA%D1%8D%D0%BC%D0%BF%D0%B0_%E2%80%94_%D0%9C%D1%8D%D1%81%D1%81%D0%B8) по небольшому отрезку ЛРП этого РСЛОС. Поэтому, для построения криптографических генераторов использует сочетание РСЛОС с различными узлами и элементами памяти. Одним из вариантов такого сочетания является комбинирующий генератор. Дополнительную информацию вы можете посмотреть в лекции.

*Комбинирующий генератор.* Криптографически стойкий [комбинирующий генератор](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80#%D0%9D%D0%B5%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BE%D0%BC%D0%B1%D0%B8%D0%BD%D0%B0%D1%86%D0%B8%D1%8F_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BE%D0%B2) построен на основе РСЛОС над полем , длины которых попарно взаимно-простые числа, и нелинейной [булевой функции](https://ru.wikipedia.org/wiki/%D0%91%D1%83%D0%BB%D0%B5%D0%B2%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) , на вход которой поступают элементы ЛРП, вырабатываемые РСЛОС. Таким образом, -ый элемент выходной последовательности, формируемой комбинирующим генератором, может быть вычислен как , где – -ый элемент ЛРП, вырабатываемой -ым РСЛОС. Важной характеристикой комбинирующего генератора, является [линейная сложность](https://chilikov.wordpress.com/2012/10/31/%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F-%D1%81%D0%BB%D0%BE%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C-%D0%BA%D0%BE%D0%BC%D0%B1%D0%B8%D0%BD%D0%B8%D1%80%D1%83%D1%8E%D1%89%D0%B8%D1%85-%D0%B3%D0%B5/). Высокую линейную сложность имеет, например, [генератор Геффе](https://ru.bmstu.wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%93%D0%B5%D1%84%D1%84%D0%B5).

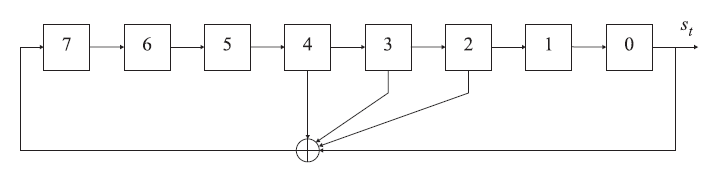
*Генератор Геффе.* Этот генератор один из вариантов комбинирующего генератора при . В нем -ый элемент выходной последовательности может быть вычислен как , где – -ый элемент ЛРП, вырабатываемой -ым РСЛОС. Для достижения максимального периода выходной последовательности, длины РСЛОС должны быть попарно взаимно-простыми числами. Но даже в этом случае данный генератор уязвим к [корреляционным атакам](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80#%D0%9A%D0%BE%D1%80%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B0%D1%82%D0%B0%D0%BA%D0%B8).

*Тестирование последовательностей.* Для проверки статистических свойств последовательностей, которые вырабатывают аппаратные и программные [генераторы псевдослучайных чисел](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D1%87%D0%B8%D1%81%D0%B5%D0%BB) используются пакеты (батареи) [статистических тестов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D1%85_%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9#%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D1%8B). Один из самых известных пакетов – пакет тестов [NIST](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D1%8B_NIST).

*Условие лабораторной работы.* В рамках данной лабораторной работы вам необходимо реализовать генератор Геффе с заданными характеристическими многочленами и начальными состояниями.

Шаг 1. Реализовать работу РСЛОС. На входе заданы число *n* – количество ячеек памяти, из которых состоит регистр, и две битовые последовательности: 1) *an*-1*an*-2...*a*1*a*0 – задает начальное заполнение, 2) *cn*-1*cn*-2...*c*1*c*0 – задает линейную обратную связь. *a*0 – задает состояние 0-ой ячейки памяти, *a*1 – перовой и т.д., *an*-1 – (*n*-1)-ой, таким образом задаем состояния всех *n* ячеек. Пусть *b*0, *b*1, ..., *bn*-1 – текущие значения в соответствующих ячейках памяти (в начальный момент *bi* = *ai*). За одну итерацию: 1) внешний выводе: регистр возвращает значение, хранящееся в 0-ой ячейке, т.е. *b*0; 2) изменение внутреннего состояния: вычисляется значение *r* = (*c*0\**b*0) + (*c*1\**b*1) +...+ (*cn*-1\**bn*-1); значение *i*-ой ячейки переходит в (*i*-1)-ую (*i* изменяется от 1 до *n*-1), т.е. *bi*-1 = *bi*; значение (*n*-1)-ой принимается равным *bn*-1 = *r*. На рисунке внизу представлен пример, для n = 8 и *c7c6*...*c*1*c*0=00011101.

На первом шаге необходимо реализовать 3 РСЛОС, заданные в вашем варианте. Для каждого регистра найти период выходной последовательности и сгенерированную последовательность, до начала зацикливания.



Шаг 2. Сгенерировать выходную последовательность *γi* генератора Геффе длительностью N = 10 000 элементов.

Шаг 3. Для сгенерированной на Шаге 2 последовательности вычислить следующие статистики:  
1) количество 0 и количество 1; 2)  для *i* от 1 до 5. Прокомментировать полученные результаты.

*Варианты РСЛОС в лабораторной работе:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Вариант** | **РСЛОС №1**  **(5)** | **РСЛОС №2**  **(7)** | **РСЛОС №3**  **(8)** |
| 1 | 01101  10111 | 1101111  1100001 | 00000011  11001101 |
| 2 | 01001  11011 | 0011100  0101111 | 10001011  00001111 |
| 3 | 10010  10011 | 0000001  0110011 | 00100101  11110011 |
| 4 | 01000  11101 | 0100001  1001111 | 01101011  00101011 |
| 5 | 00010  01111 | 1100101  0101111 | 00000001  10011101 |
| 6 | 00111  10011 | 1100000  1101011 | 11010100  11010101 |
| 7 | 00101  10101 | 0111001  1100111 | 10010100  11010011 |
| 8 | 10100  11001 | 1011000  1000011 | 01000110  00001101 |
| 9 | 10010  00111 | 1011011  0110011 | 00110010  11010011 |
| 10 | 01101  01111 | 1000010  1000111 | 00110010  01001101 |
| 11 | 00100  01111 | 1010111  1010001 | 00101100  11110101 |
| 12 | 11100  10111 | 1110101  1101011 | 10010111  10111011 |
| 13 | 10010  01101 | 0010000  0101101 | 01100011  11110011 |
| 14 | 00100  10111 | 0111010  0011001 | 00000100  11100011 |
| 15 | 01110  11101 | 1110101  0011011 | 01111110  10101111 |
| 16 | 00000  10111 | 1110110  0111011 | 00001100  11100101 |
| 17 | 11111  10111 | 0111110  0011001 | 11101011  11000011 |
| 18 | 00110  01111 | 0011011  1011011 | 11001101  10010111 |
| 19 | 01011  01101 | 0000101  0111011 | 10011011  11000101 |
| 20 | 11010  10101 | 1010000  0101101 | 00111110  00001111 |
| 21 | 10100  11011 | 1000110  1001101 | 11000000  00001011 |
| 22 | 01010  10101 | 0010100  0111111 | 10000001  01101001 |
| 23 | 11111  01011 | 1101100  1011001 | 00011011  00011111 |