Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики

Лабораторная работа № 3

по дисциплине «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

ГЕНЕРАЦИЯ КРИПТОГРАФИЧЕСКИ БЕЗОПАСНОЙ ПСЕВДОСЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

Факультет:	ПМИ
Группа:	ПМИ-02
Вариант:	6
Студенты:	Сидоров Даниил,
	Дюков Богдан
Преподаватели:	Авдеенко Татьяна Владимировна,
	Кутузова Ирина Александровна.

Новосибирск

2026

1. Цель работы

Изучить алгоритмы и получить навыки генерации криптографически безопасной псевдослучайной последовательности. Научиться тестировать полученную последовательность на равномерность и случайность.

2. Задача

I. Реализовать приложение с графическим интерфейсом, позволяющее выполнять следующие действия.

1. Генерировать псевдослучайную последовательность с помощью заданного в варианте алгоритма:
   1) все входные параметры генератора должны задаваться из файла или вводиться в приложении;
   2) сгенерированная последовательность, состоящая из и , должна сохраняться в файл.
2. Проверять полученную псевдослучайную последовательность на равномерность и случайность с помощью трех рассмотренных тестов:
   1) результат проверки каждого теста должен отображаться в приложении;
   2) все вычисляемые промежуточные значения (все шаги алгоритма теста) могут отображаться в приложении или сохраняться в файл.

II. С помощью реализованного приложения выполнить следующие задания.

1. Протестировать правильность работы разработанного приложения.
2. Сгенерировать последовательность из не менее 10 000 бит и исследовать еѐ на равномерность и случайность.
3. Сделать вывод о случайности сгенерированной последовательности и о возможности ее использования в качестве криптографически безопасной псевдослучайной последовательности.

Вариант	Алгоритм
6	ANSI X9.17

3. Метод решения задачи

1) Алгоритм ANSI X9.17

ANSI X9.17 – алгоритм, являющийся национальным стандартом США для генерации двоичной псевдослучайной последовательности. Используется в приложениях, обеспечивающих безопасность финансовых платежей и PGP.

Входные данные: некоторое случайное (и секретное) 64-битное начальное значение s_0, 128-битный составной ключ K (включает в себя $K_1 \,||\, K_2$) и m - количество генерируемых 64-битных двоичных слов.

Выходные данные: последовательность m 64-битных двоичных слов x_1, x_2, …, x_m.

В этом генераторе в качестве односторонней функции используется алгоритм шифрования TripleDES:

$$\mathrm{F_K(M)} = \mathrm{E_{K_1}}\left(\mathrm{D_{K_2}}\left(\mathrm{E_{K_1}(M)}\right)\right),$$

где $\mathrm{E_{K_1}}(M)$ – шифрование сообщения M алгоритмом DES с ключом $\mathrm{K_1}$, $\mathrm{D_{K_2}(M)}$ – дешифрование сообщения M алгоритмом DES с ключом $\mathrm{K_2}(\mathrm{K_2} \neq \mathrm{K_1})$.

Шаги алгоритма:

1. Фиксируется 64-битное представление d даты и времени (количество 100-наносекундных интервалов, прошедших с 12:00:00 полуночи, 1 января 1 года н. э., переведенное в двоичный вид) в момент обращения к программе генерации и вычисляется вспомогательное 64-битное двоичное слово: $Temp = F_k(d)$.
2. Для $i = \overline{1,m}$:
   1) вычисляется значение i-го выходного слова $x_i = F_k(Temp \oplus s_{i-1})$;
   2) вычисляется новое значение параметра $s_i = F_k(x_i \oplus Temp)$.
3. В результате предыдущего шага формируется выходная псевдослучайная последовательность из m слов x_1, x_2, …, x_m (либо двоичная псевдослучайная последовательность из $64 * m$ бит: $X = x_1 \,||\, x_2 \,||\, \ldots \,||\, x_m$).

Алгоритм шифрования DES не реализовывался самостоятельно (использовалась готовая реализация).

2) **Статистические тесты**

В данной лабораторной работе рассматривается наборов из 3 тестов, оценивающих «случайность» псевдослучайной последовательности.

Частотный тест

Это статистический тест, который используется для проверки случайности бинарной последовательности. Он оценивает равномерность распределения нулей и единиц в последовательности. Кратко алгоритм:

1) **Преобразование последовательности**. Входная последовательность, состоящая из 0 и 1, преобразуется в последовательность -1 и 1.
2) **Вычисление суммы** элементов преобразованной последовательности.
3) **Вычисление статистики**, которая представляет собой абсолютное значение суммы, деленное на квадратный корень из количества элементов в последовательности.
4) **Оценка статистики**. Если значение статистики меньше или равно 1.82138636 (критическое значением для статистики при уровне значимости 0.01), то

последовательность считается случайной (нули и единицы равномерно распределены).

Если последовательность не проходит частотный тест, то нет необходимости проводить дальнейшие тесты, поскольку уже ясно, что она не является равномерно распределенной.

Тест на последовательность одинаковых бит

Это статистический тест, который анализирует количество непрерывных последовательностей (цепочек) одинаковых бит в проверяемой последовательности. Кратко алгоритм:

1) **Вычисление частоты единиц**. Вычисляется частота, с которой в проверяемой последовательности встречаются единицы.
2) **Вычисление количества цепочек**. Вычисляется значение V_n, которое представляет собой количество цепочек в последовательности.
3) **Вычисление статистики**, которая представляет собой нормализованное отклонение значения V_n от его ожидаемого значения для истинно случайной последовательности.
4) **Оценка статистики**. Если значение статистики меньше или равно 1.82138636, то последовательность считается случайной.

Тест определяет, является ли количество цепочек из нулей и единиц различной длины в последовательности приблизительно таким же, как должно быть в истинно случайной последовательности.

Расширенный тест на произвольные отклонения

Это статистический тест, который оценивает общее число посещений определённого состояния при произвольном обходе кумулятивной суммы. Вот краткое описание этого теста:

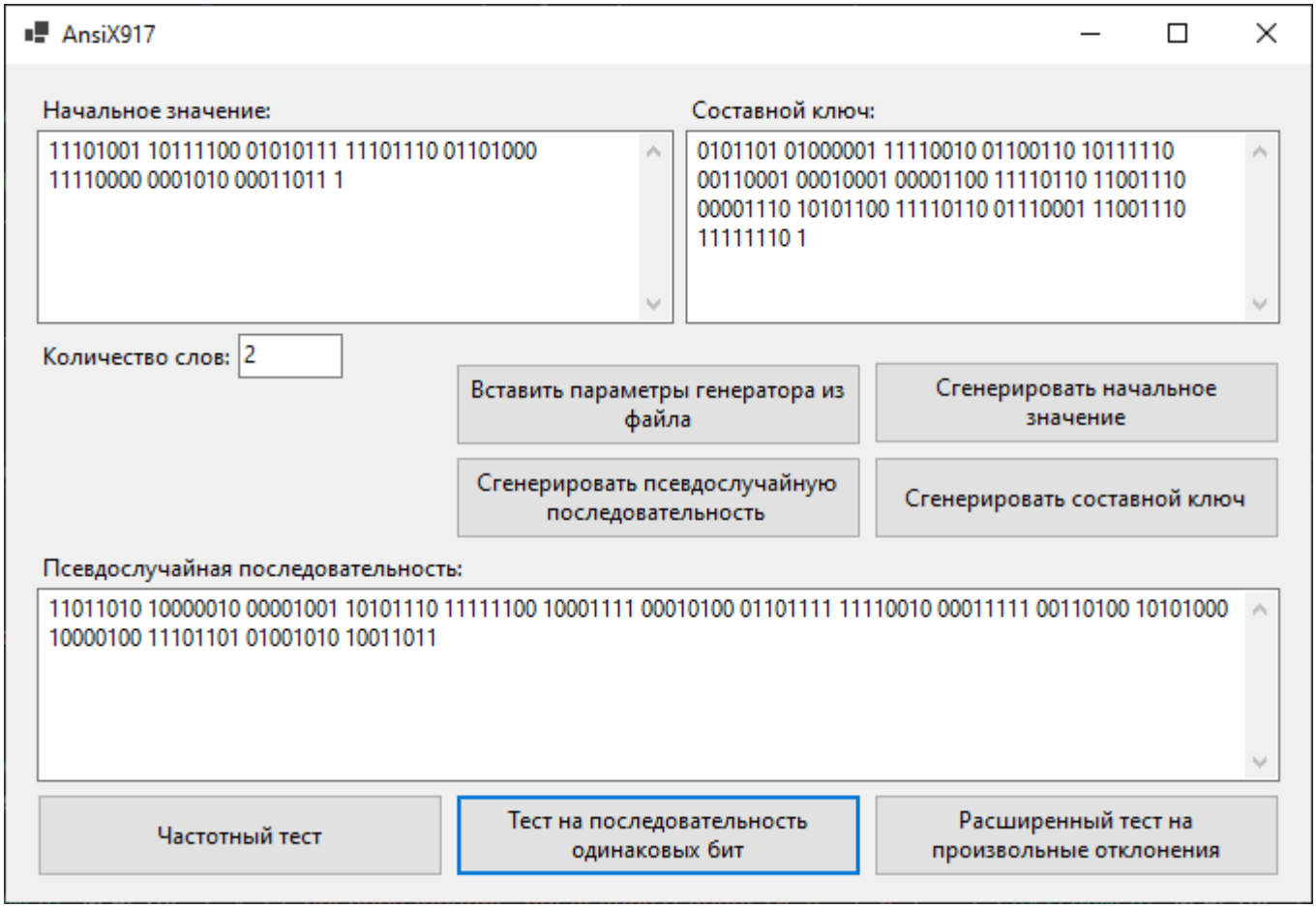
1) **Преобразование последовательности**: Входная последовательность, состоящая из 0 и 1, преобразуется в последовательность -1 и 1.
2) **Вычисление сумм** последовательно удлиняющихся подпоследовательностей, начинающихся с первого элемента.
3) **Формирование новой последовательности**. Суммы из 2 шага формируют последовательность. Новая последовательность состоит из этой последовательности, в которую в начало и конец добавляется 0.
4) **Вычисление количества нулей** в полученной последовательности минус 1.
5) **Вычисление ξj**. Для каждого из 18 состояний (-9, -8, …, -1, 1, 2, …, 9) вычисляется ξj, которое показывает, сколько раз состояние j встречалось в последовательности.
6) **Вычисление статистик** для каждого состояния (18 статистик).

7) **Оценка статистик**. Если все статистики меньше или равны 1.82138636, то последовательность считается случайной. В противном случае, последовательность считается неслучайной.

4. Разработанное программное средство

Разработанное программное средство представляет собой приложение Windows Forms.

Интерфейс главного меню:

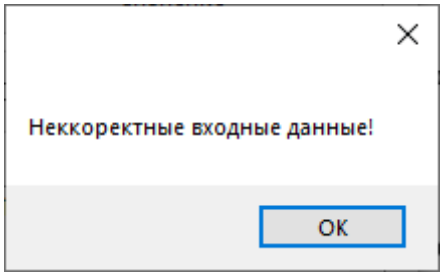


Он содержит текстовые поля для входных данных алгоритма (начальное состояние, составной ключ и количество слов в результате), а также для результирующей псевдослучайной последовательности.

Каждое текстовое поле обрабатывает корректность ввода. Для начального значения: разрешен только двоичный ввод + Backspace, длина без пробелов - 64 бита. Для составного ключа: разрешен только двоичный ввод + Backspace, длина без пробелов – 128 бит. Для количества слов: можно вводить только цифры + Backspace, запрещено начинать число с 0. Для псевдослучайной последовательности: разрешен только двоичный ввод + Backspace.

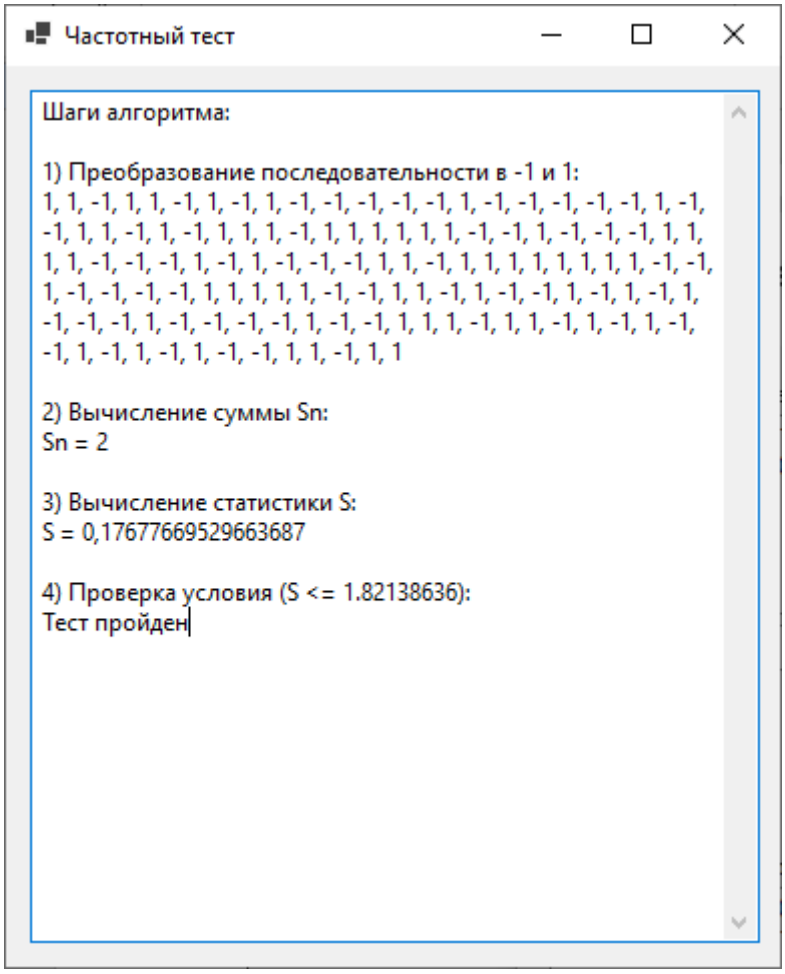
Имеется возможность вставить входные данные из файла, а также сгенерировать их случайно (сгенерировать начальное состояние или составной ключ). После нажатия на кнопку “Сгенерировать псевдослучайную

последовательность”, генерируется псевдослучайная последовательность только в том случае, если все входные данные удовлетворяют условиям. Иначе генерация не выполняется и выбрасывается сообщение:

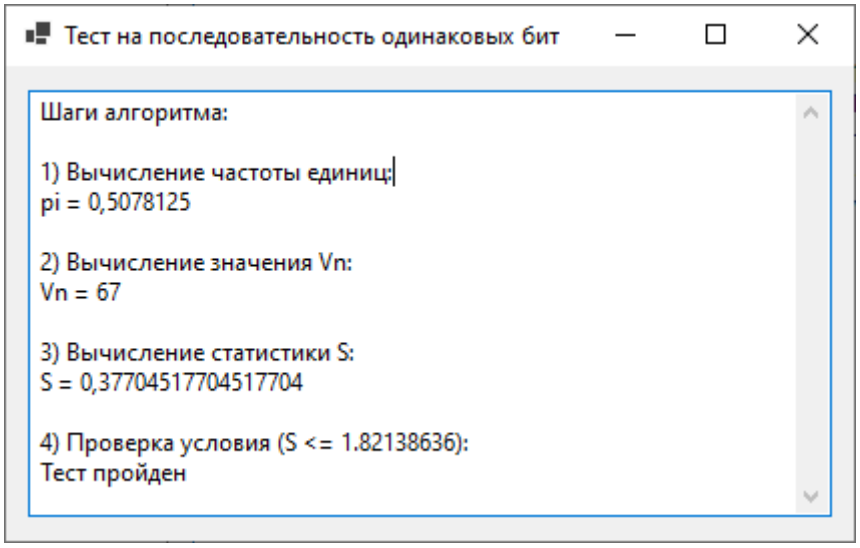


Когда псевдослучайная последовательность сгенерирована, входные и выходные данные сохраняются в соответствующие файлы. Также имеется возможность оценить её равномерность и случайность с помощью трех тестов. Когда нажимается кнопка, соответствующая любому тесту, генерируется отдельная форма с информацией о шагах алгоритма теста и его результатах (дополнительно эта же информация сохраняется в соответствующий файл).

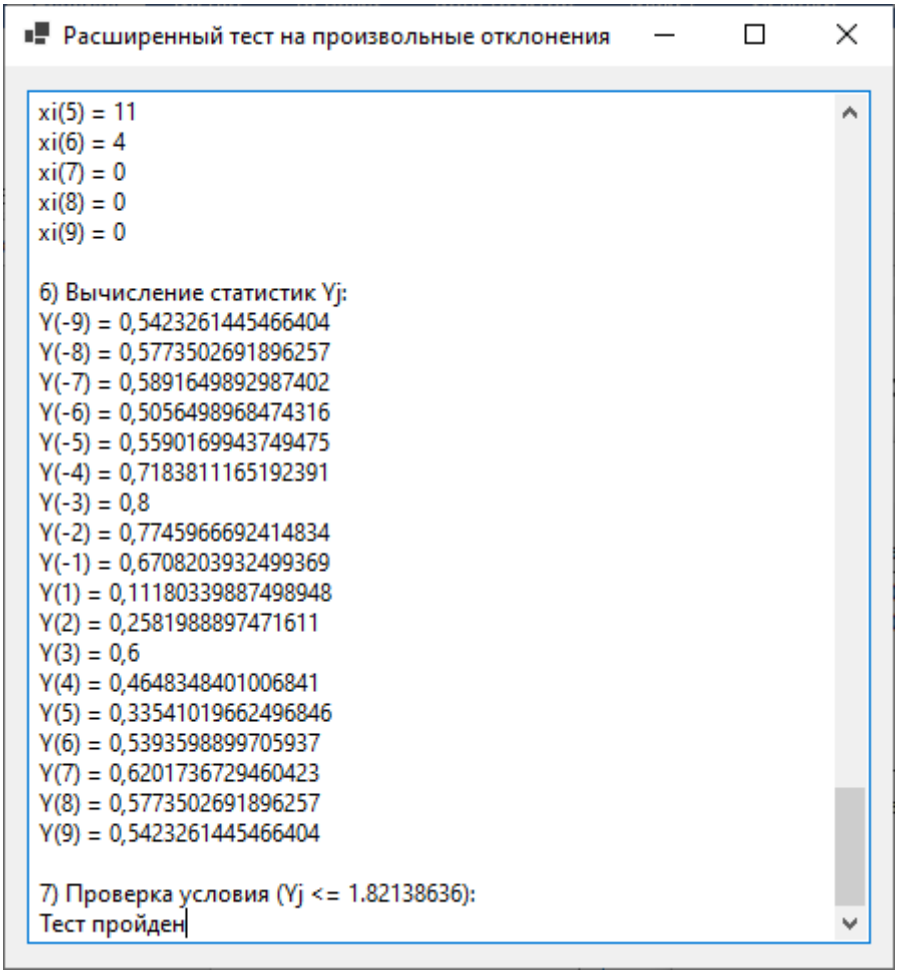
Форма с результатами частотного теста:



Форма с результатами теста на последовательность одинаковых бит:



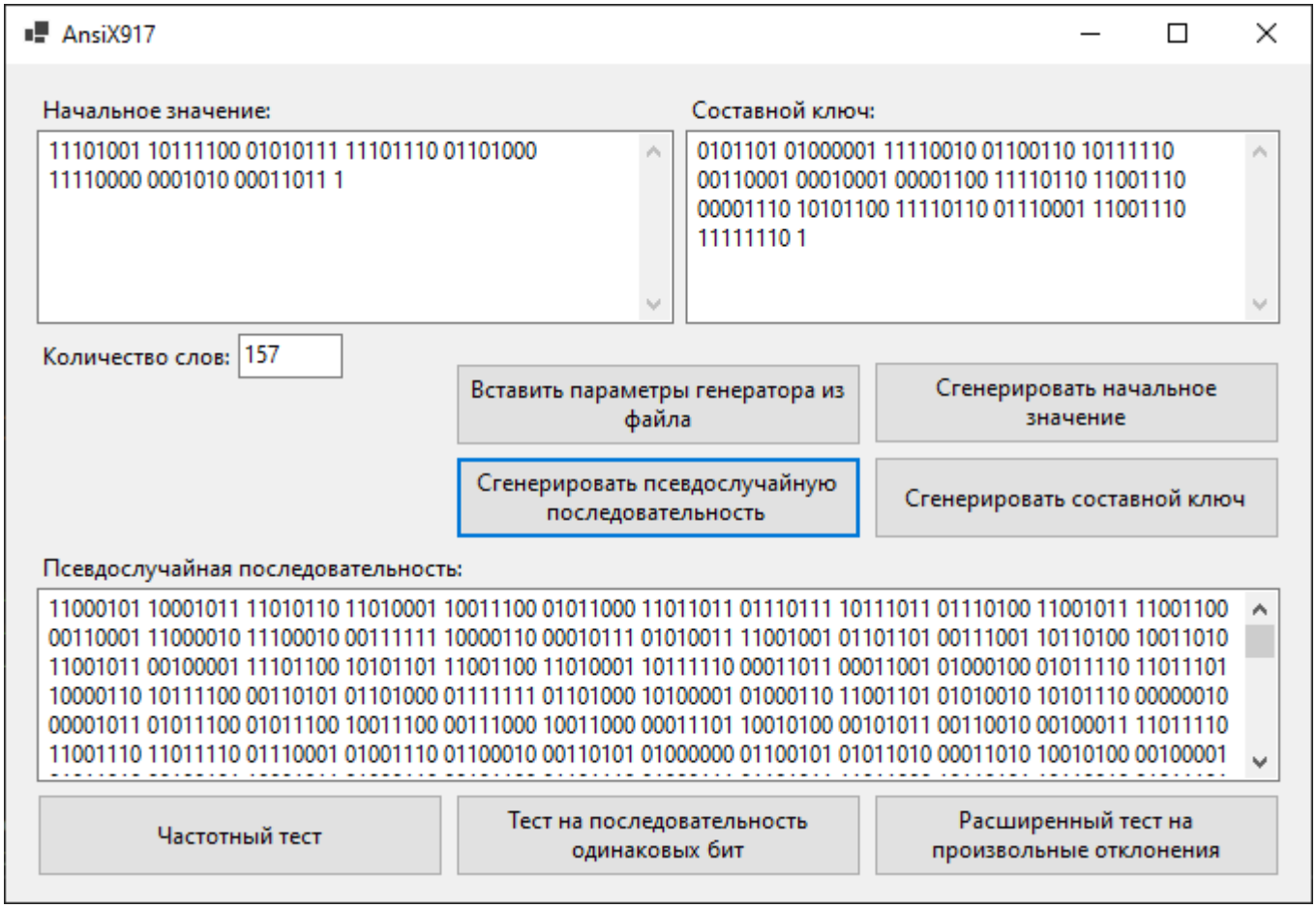
Форма с результатами расширенного теста на произвольные отклонения:



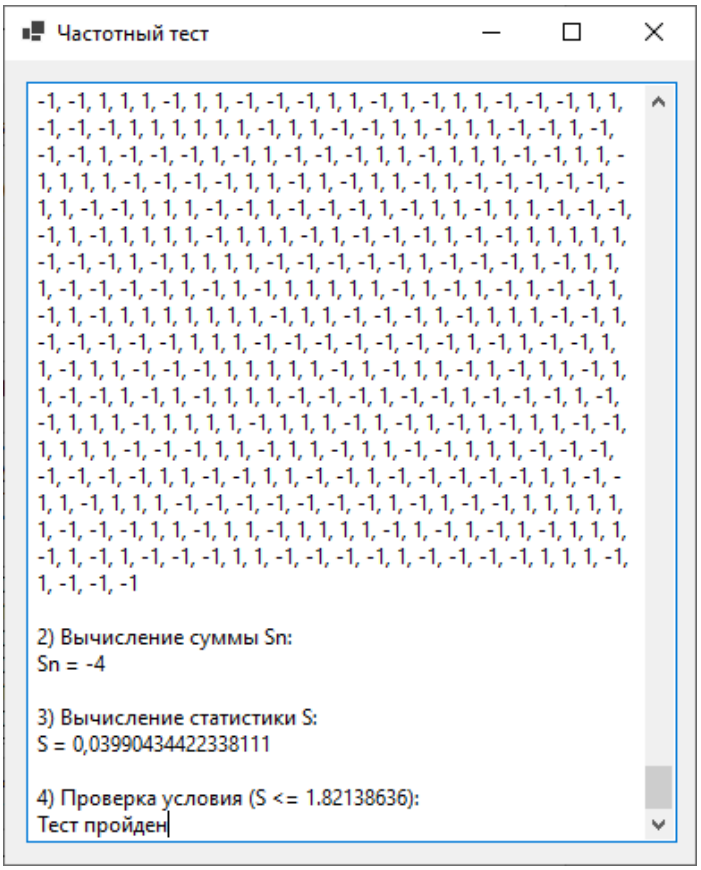
5. Тестирование

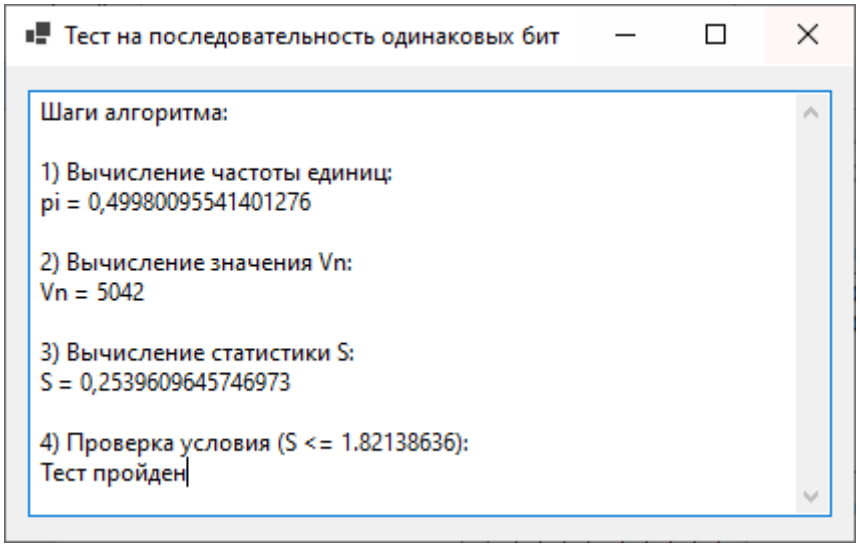
Демонстрационный тест из прошлого пункта показывает корректность работы алгоритма генерации. Сгенерировалось действительно 2 64-битных слова. Если мы попробуем сгенерировать ещё раз, то последовательность будет иной, что также подчеркивает правильность алгоритма, так как в нем используется текущие значения даты и времени.

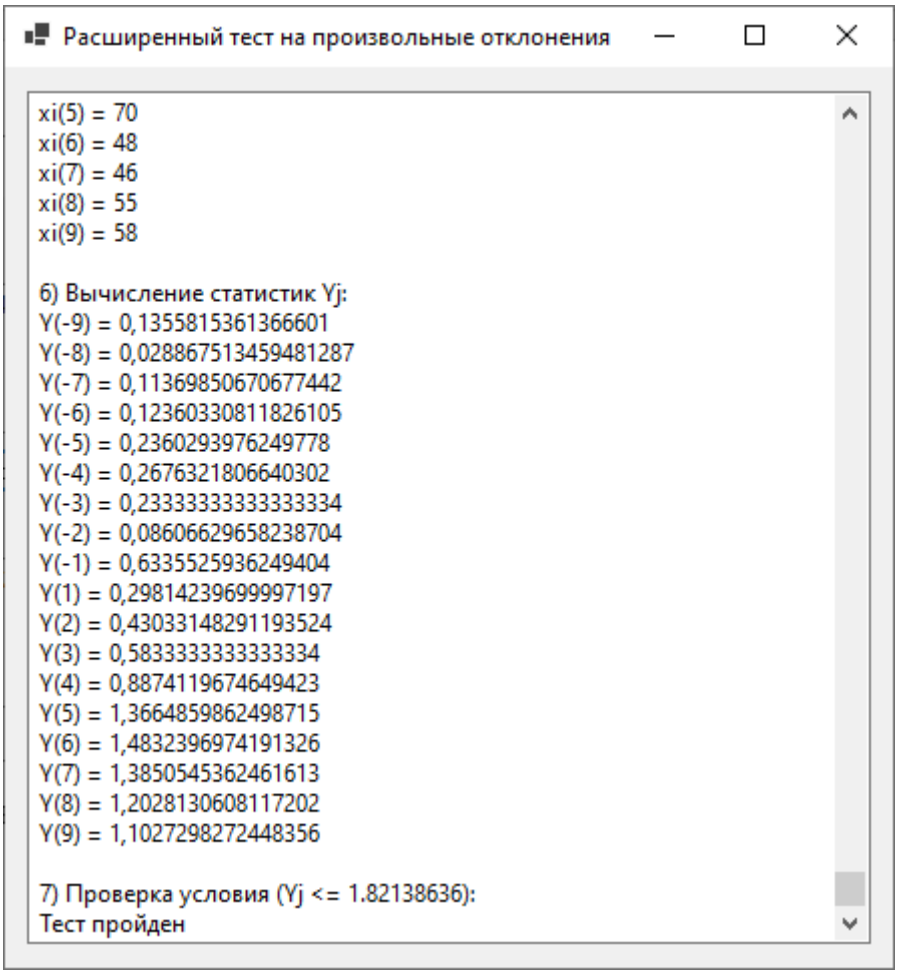
Сгенерируем последовательность из 10 048 бит и исследуем её на равномерность и случайность:



Результаты всех тестов:







6. Вывод

В ходе проведения лабораторной работы, мы освоили алгоритм генерации псевдослучайной последовательности ANSI X9.17, в котором используется алгоритм шифрования TripleDES в качестве односторонней функции.

Для анализа случайности сгенерированной последовательности (в данном случае длиной более 10000 бит) и её возможности использования в качестве криптографически безопасной псевдослучайной последовательности, мы провели статистические тесты.

Частотный тест помог оценить пропорцию нулей и единиц в проверяемой последовательности. Количество нулей и единиц в последовательности приблизительно одинаково, поэтому тест пройден.

Тест на последовательность одинаковых бит помог проанализировать количество цепочек в проверяемой последовательности, где цепочка – это непрерывная последовательность одинаковых бит. Количество цепочек из нулей и единиц различной длины в последовательности получилось приблизительно такое же, как должно быть в истинно случайной последовательности, поэтому она проходит этот тест.

Расширенный тест на произвольные отклонение помог оценить общее число посещений определённого состояния при произвольном обходе кумулятивной суммы. Все 18 статистик $Y_j \leq 1.82138636$, поэтому тест считается успешно пройденным.

Таким образом, сгенерированная последовательность является равномерно распределенной и случайной, и, следовательно, может быть использована в качестве криптографически безопасной псевдослучайной последовательности.