Лабораторная работа №1

Изучение генераторов псевдослучайных последовательностей

Задание:

1. Изучить алгоритм работы генераторов псевдослучайных чисел (ГПЧ)
2. Реализовать приложение, которое позволяет генерировать псевдослучайную последовательность чисел четырьмя различными способами:
    - С помощью генератора встроенного в язык программирования высокого уровня
    - С помощью встроенного криптографического генератора в криптопровайдер Windows
    - С помощью первого генератора реализованного согласно варианту задания (Таблица №1)
    - С помощью второго генератора реализованного согласно варианту задания (Таблица №1)
3. Исследовать полученную последовательность на случайность с помощью методики NIST STS
4. Подготовить и оформить отчет о проведенной лабораторной работе.

Дополнительные требования к приложению:

1. Пользователь имеет возможность выбирать одну из четырех схем генерации псевдослучайных чисел.
2. Должна быть возможность задавать начальное значение для работы генератора, а также размер генерируемой последовательности в байтах
3. Программа должна быть оформлена в виде удобной утилиты, позволяющей сохранять полученную последовательность в файл
4. Текст программы оформляется прилично (удобочитаемо, с описанием ВСЕХ функций, переменных и критических мест).
5. В процессе работы программа ОБЯЗАТЕЛЬНО выдает информацию о состоянии процесса формирования последовательности
6. Интерфейс программы может быть произвольным, но удобным и понятным (разрешается использование библиотек VCL)
7. Среда разработки и язык программирования могут быть произвольными.

Порядок проведения анализа полученной последовательности:

1. Запустить приложение NIST_STS.exe для статистического тестирования
2. Выполнить статистическое тестирование полученных данных по каждому генератору. Для этого необходимо воспользоваться руководством Приложение А.
3. Используя обобщенные результаты тестирования, которые находятся в файле finalAnalysisReport, выполнить интерпретацию результатов тестирования (См. Описание тестов в Приложение Б).

Требования для сдачи лабораторной работы:

1. Демонстрация работы реализованной вами системы.
2. АВТОРСТВО
3. Теория
4. Оформление и представление письменного отчета по лабораторной работе, который содержит:
    1. Титульный лист

2. Задание на лабораторную работу
3. Описание используемых алгоритмов реализации генераторов
4. Листинг программы
5. Отчет по результатам анализа по каждому генератору и по каждому тесту

Варианты заданий.

Таблица 1.

№ варианта	Схема генератора псевдослучайной последовательности А
1.	Линейный конгруэнтный генератор + Генератор Геффа
2.	Квадратичный конгруэнтный генератор + Аддитивный генератор
3.	Кубический конгруэнтный генератор + Генератор Стоп-Пошел
4.	Генератор Геффа + Квадратичный конгруэнтный генератор
5.	Генератор Стоп-Пошел + Генератор Парка-Миллера
6.	Аддитивный генератор + Кубический конгруэнтный генератор
7.	Генератор Парка-Миллера + Генератор Геффа
8.	Линейный конгруэнтный генератор + Аддитивный генератор
9.	Квадратичный конгруэнтный генератор + Генератор Стоп-Пошел
10.	Кубический конгруэнтный генератор + Генератор Геффа
11.	Генератор Геффа + Генератор Стоп-Пошел
12.	Генератор Стоп-Пошел + Аддитивный генератор
13.	Аддитивный генератор + Генератор Парка-Миллера
14.	Генератор Парка-Миллера + Генератор Геффа
15.	Линейный конгруэнтный генератор + Аддитивный генератор
16.	Квадратичный конгруэнтный генератор + Генератор Геффа
17.	Кубический конгруэнтный генератор + Аддитивный генератор
18.	Генератор Геффа + Аддитивный генератор
19.	Генератор Стоп-Пошел + Квадратичный конгруэнтный генератор
20.	Аддитивный генератор + Генератор Парка-Миллера
21.	Генератор Парка-Миллера + Генератор Стоп-Пошел
22.	Линейный конгруэнтный генератор + Генератор Геффа
23.	Квадратичный конгруэнтный генератор + Аддитивный генератор
24.	Кубический конгруэнтный генератор + Аддитивный генератор
25.	Генератор Геффа + Квадратичный конгруэнтный генератор
26.	Генератор Стоп-Пошел + Генератор Парка-Миллера
27.	Аддитивный генератор + Линейный конгруэнтный генератор
28.	Генератор Парка-Миллера + Генератор Стоп-Пошел
29.	Линейный конгруэнтный генератор + Генератор Стоп-Пошел
30.	Квадратичный конгруэнтный генератор + Генератор Геффа

Приложение А. Использование программы NIST STS для анализа полученных последовательностей

Средством тестирования NIST STS следует пользоваться из командной строки. Для этого необходимо произвести запуск исполняемого файла NIST_STS.exe, и длины исследуемой последовательности. Например необходимо исследовать файл размером 12 Мб (12 582 912 байт), что составит 100 663 296 бит (такая длина позволяет говорить о 100 последовательностях длиной 1 000 000 бит, допускаются вариации на тему разложения числа 100 000 000 на два множителя: количество двоичных последовательностей и их двоичной длины), см. рис. 2.

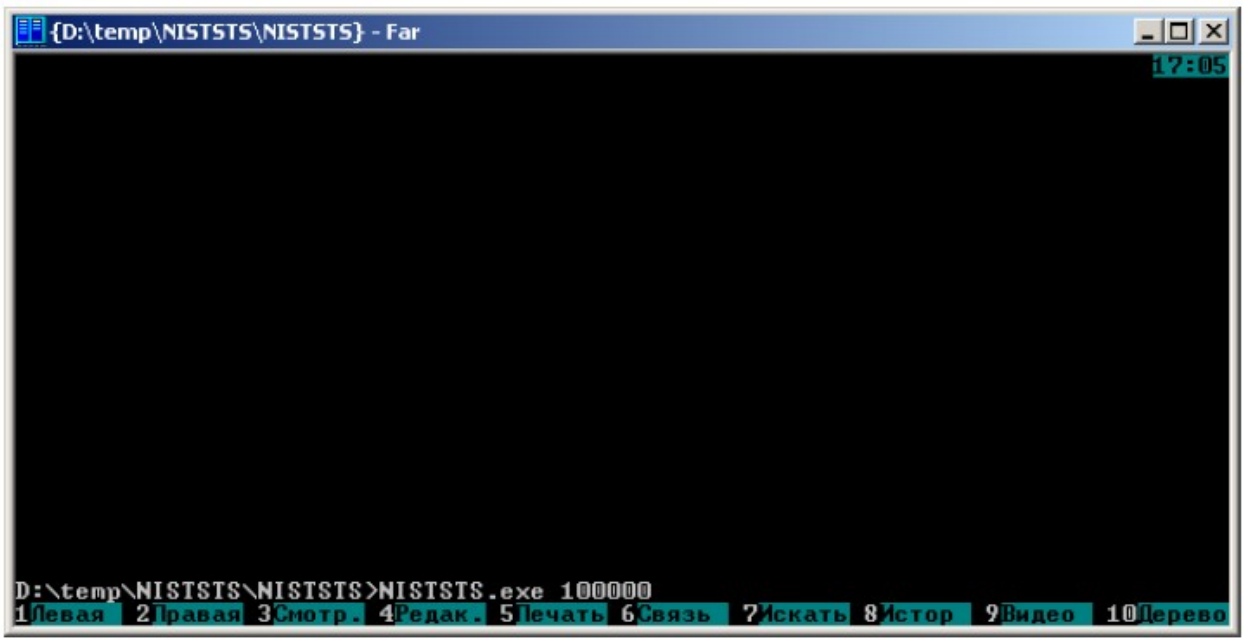


Рисунок 1: Запуск приложения NIST STS

После запуска приложения необходимо указать, какие данные необходимо тестировать, «OPTION---->»:

- Тестовые данные будут сформированы генератором, реализованным в тесте NISTSTS.exe. Для этого необходимо ввести значение от 1 до 9. Например вводим 9 для выбора генератора основанного на функции хеширования SHA-1 («G Using SHA-1»). После чего будет сформирована последовательность длиной 12 Мб.
- Тестовые данные находятся в файле, который содержит последовательность, которую необходимо протестировать. Для этого необходимо указать значение 0. В строке «User Prescribed Input File: » вводится имя файла, например «output.bin», Рис. 3.

После чего будет предоставлен перечень статистических тестов, которые могут быть применены к тестируемой последовательности. Если исследователь не желает применить существующие тесты к исследуемой последовательности, ему следует ввести в строке «Enter Choise: » цифру 0, в противном случае 1, рис. 4 и нажать клавишу «Enter».

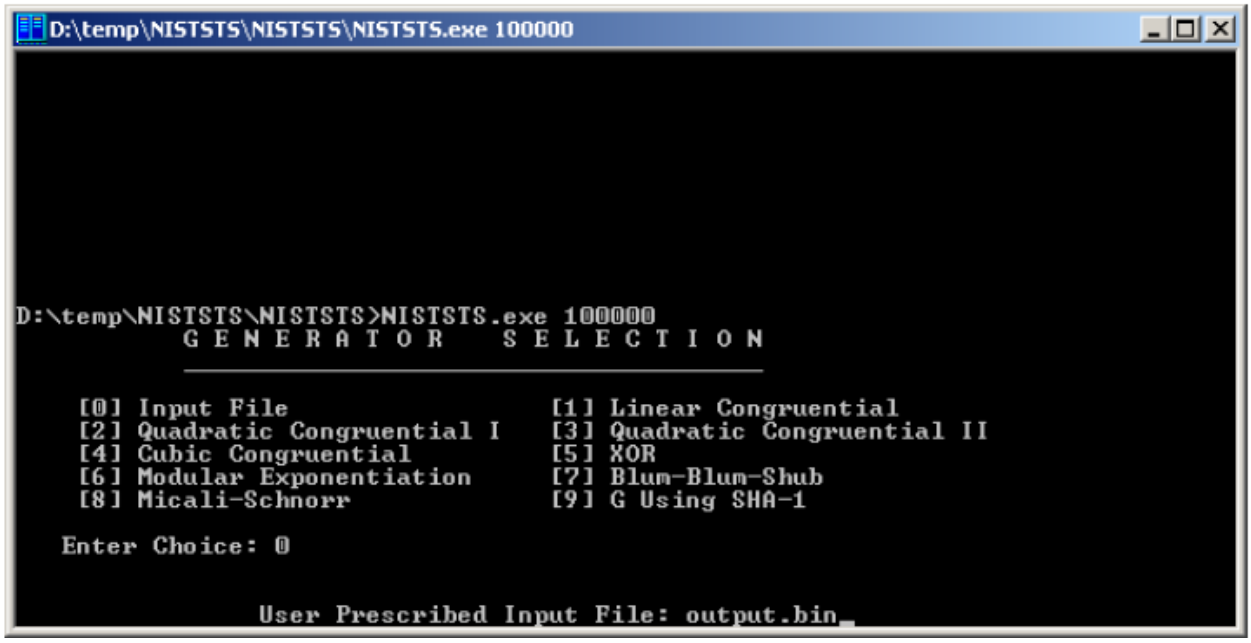


Рисунок 2: Выбор тестируемой последовательности

Далее программа предложит выбрать, какие именно тесты следует применить к последовательности. Все присутствующие тесты пронумерованы от 01 до 15. Исследователю следует ввести в нижней строке под номером соответствующего теста цифру 1, если тест следует применить, в противном случае – цифру 0, рис. 4.

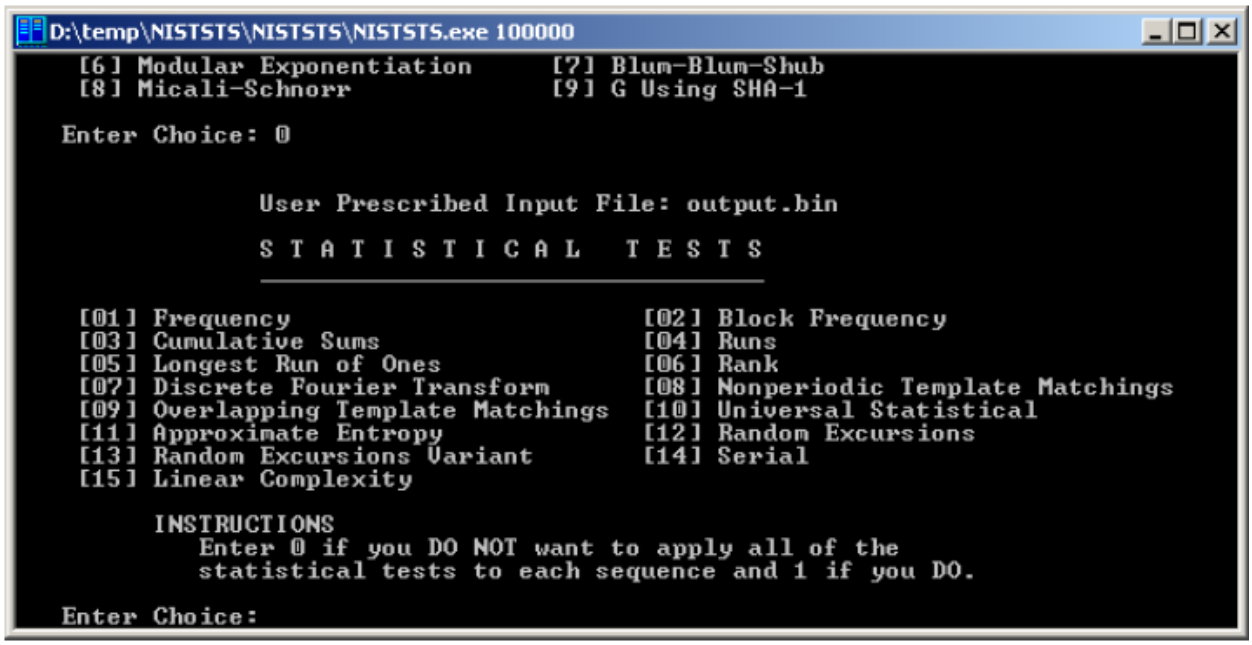


Рисунок 3: Выбор единственного статистического теста

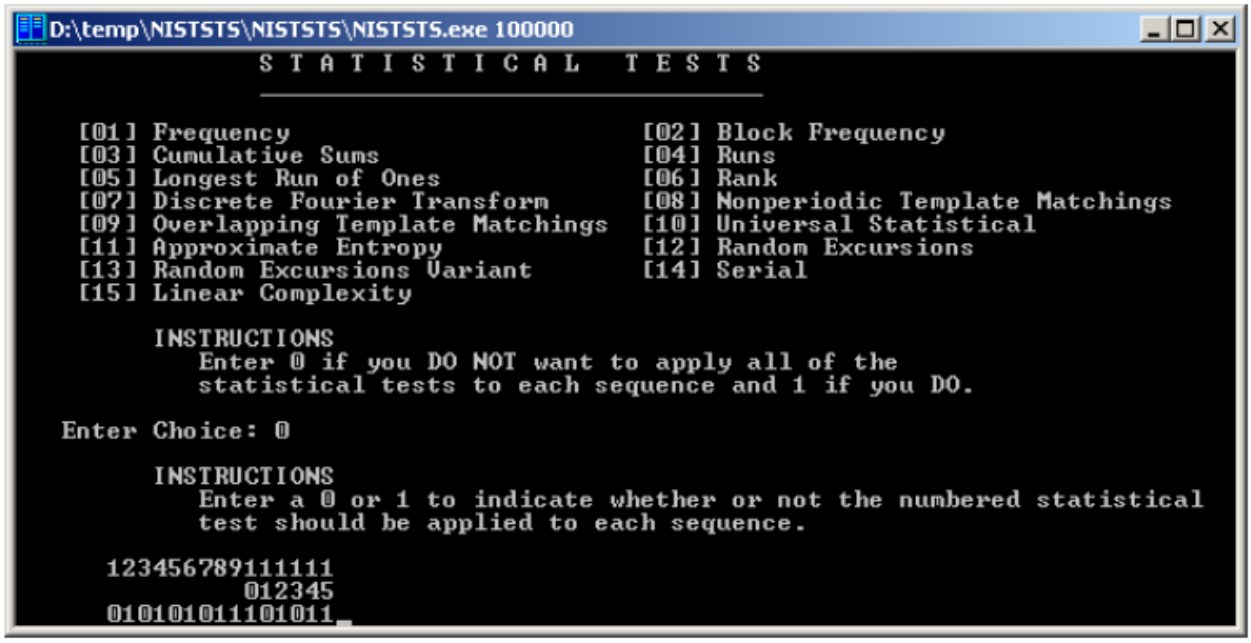


Рисунок 4: Выбор множества статистических тестов, которые будут последовательноприменяться к исследуемой последовательности

После выбора необходимых тестов, приложение предложит ввести параметры, согласно которым будет производиться тестирование последовательности. Обратим внимание, что существуют параметризированные и непараметризированные тесты. К непараметризируемым тестам следует отнести:

- Cumulative Sums
- Runs
- Longest Runs of Ones
- Rank
- Spectral DFT
- Random Excursions Variant
- Lempel Ziv Complexity
- Frequency

Для указанных тестов необходимо указать лишь длину последовательности и их количество. К параметризованным тестам следует отнести (укажем также их параметры):

- Block Frequency – длина блока, по умолчанию – 128 бит.
- NonOverlapping Template – длина блока, по умолчанию – 9 бит.
- Overlapping Template – длина блока, по умолчанию – 9 бит.
- Approximate entropy – длина блока, по умолчанию – 10 бит.
- Serial – длина блока, по умолчанию – 16 бит.
- Linear Complexity – длина блока, по умолчанию – 500 бит.

Кроме того, следует указать в каком формате будут представлены данные в файле:

- «текстовый» – один символ – один бит;
- «двоичный» - один байт содержит 8 бит последовательности.

В строке «How many bitstreams should be generated?» исследователю следует ввести количество тестируемых последовательностей. NIST STS рекомендует число 100 в качестве количества подпоследовательностей.

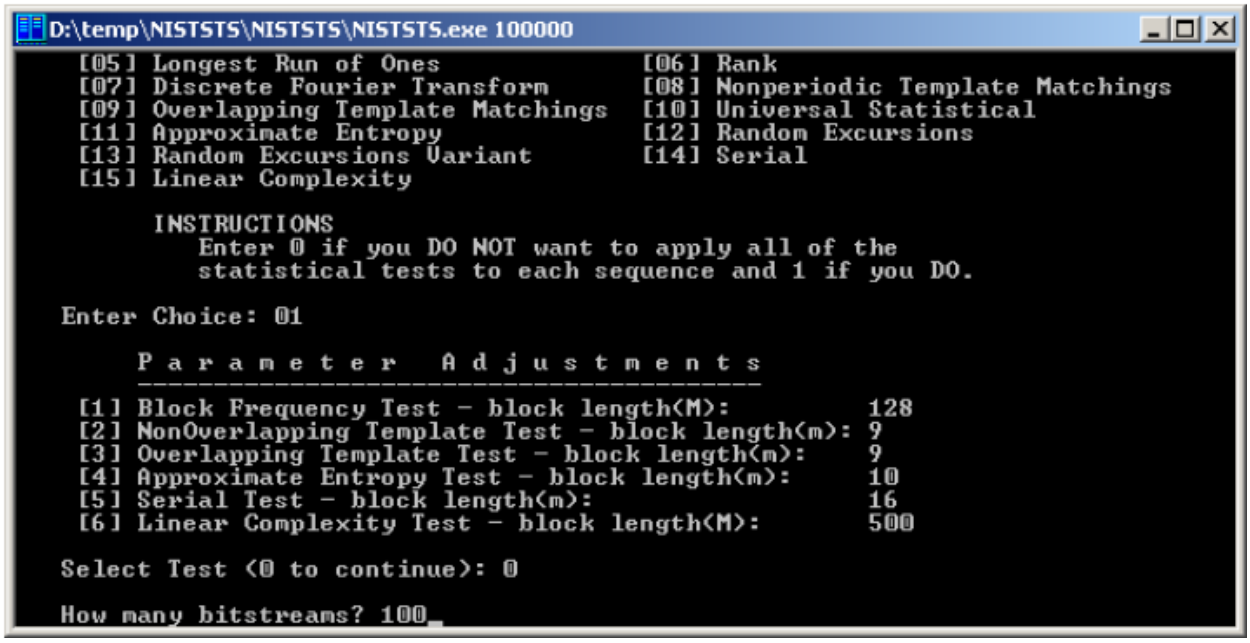


Рисунок 5: Настройка параметров Частотного теста

В строке «Select input mode:» исследователь указывает тип тестируемой подпоследовательности: в случае если последовательность представлена в ASCII формате, следует ввести 0, если в двоичном, то 1, рис. 7.

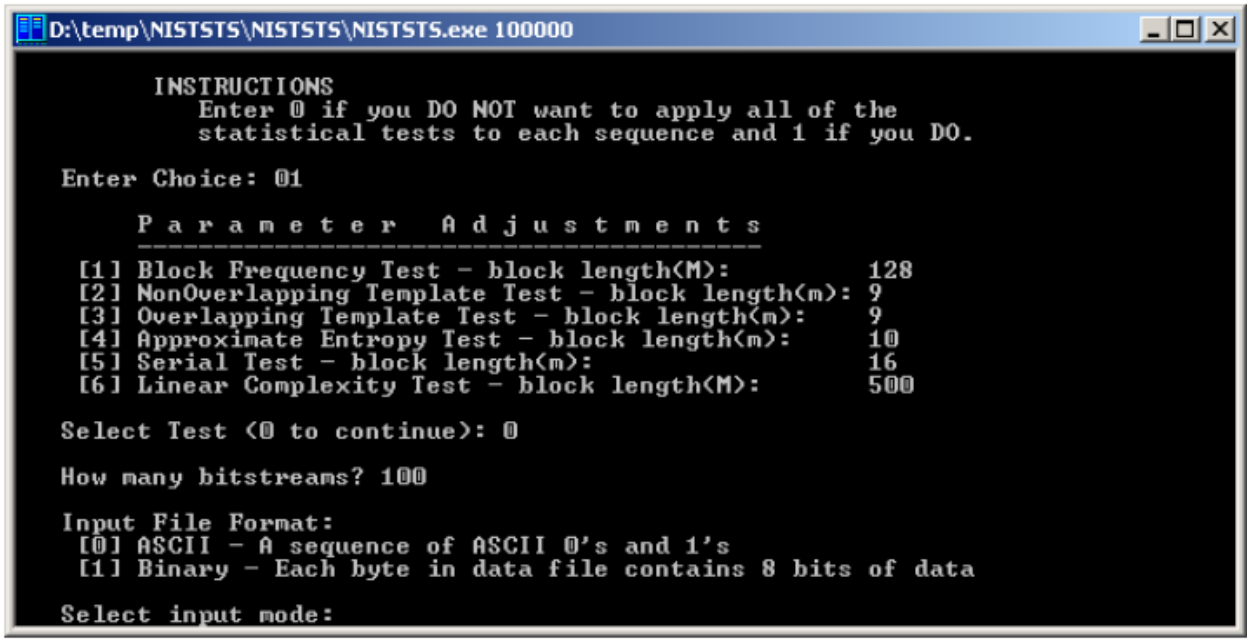


Рисунок 6: Выбор формата представления последовательности

После ввода типу последовательности приложение выведет сообщение «Statistical Testing In Progress…….», рис. 8.

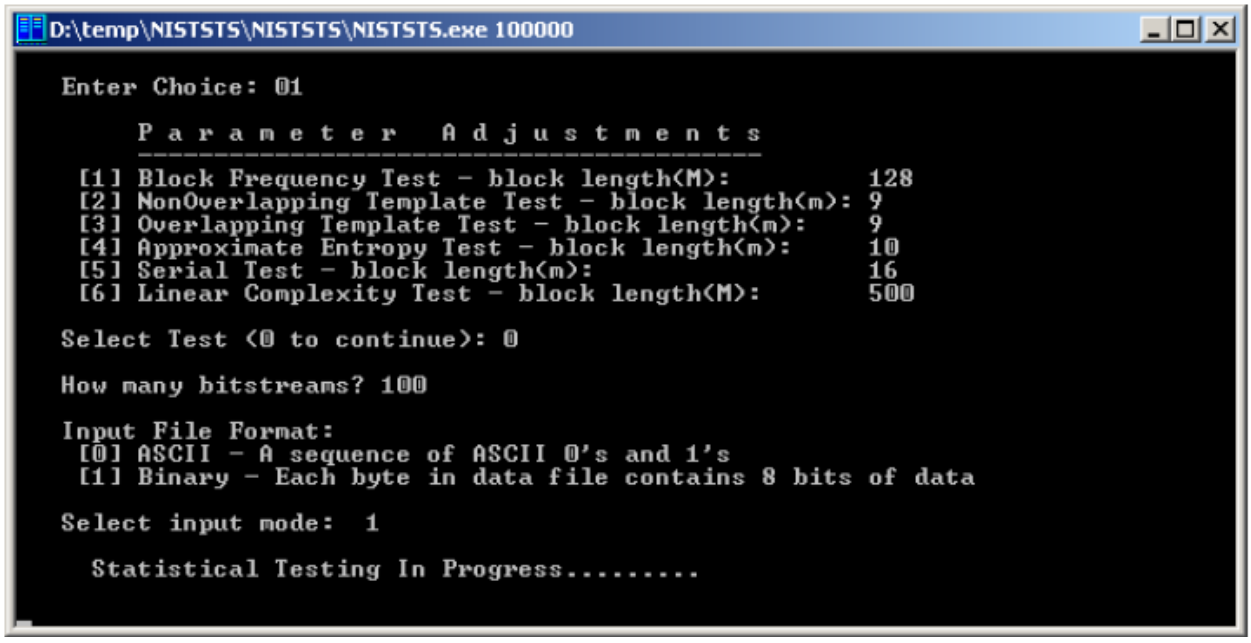


Рисунок 7: Отображение статуса теста

После завершения работы приложения, все суммарные расчетные данные размещаются в том же каталоге, где находится само приложение, в файле finalAnalysisReport, рис. 9 и рис. 10.

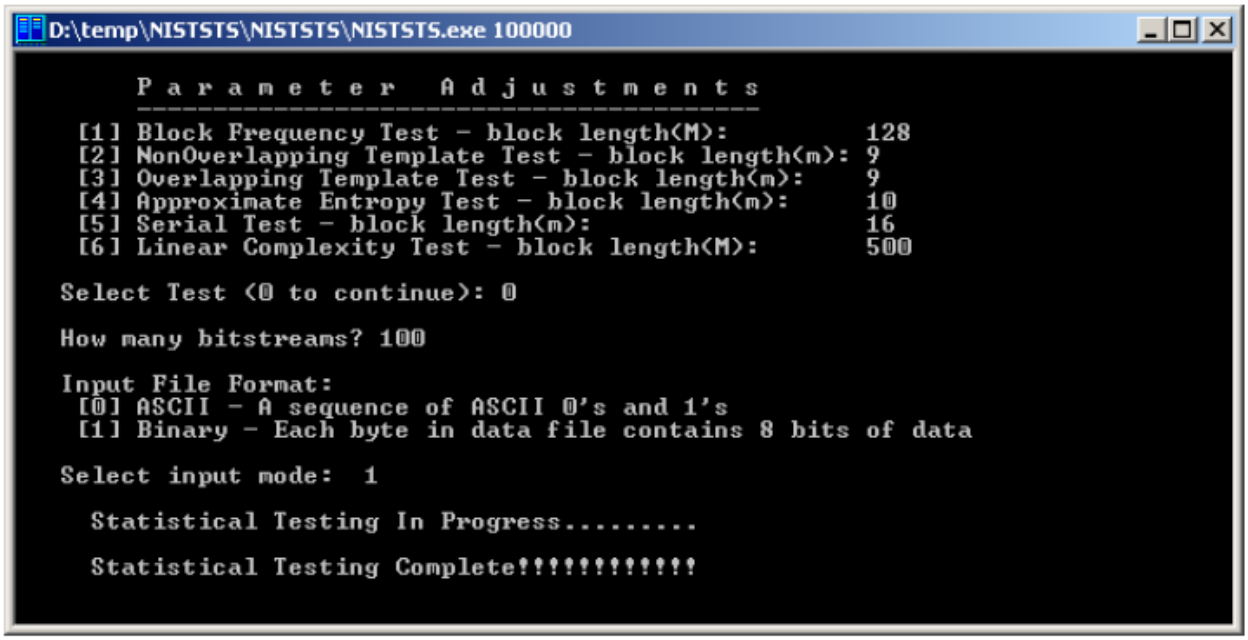


Рисунок 8: Сообщение о завершении теста

За более детальной информацией (промежуточной) следует обращаться в папку experiments, в которой перечислены папки (с названиями соответствующих тестов), будут находиться 2 файла stats и results. Файл stats содержит статистическую информацию по каждому тесту, а также формализованный результат: «Прошел» либо «Не прошел». Файл results содержит лишь значения P -вероятности, которая также указывается в файле stats.

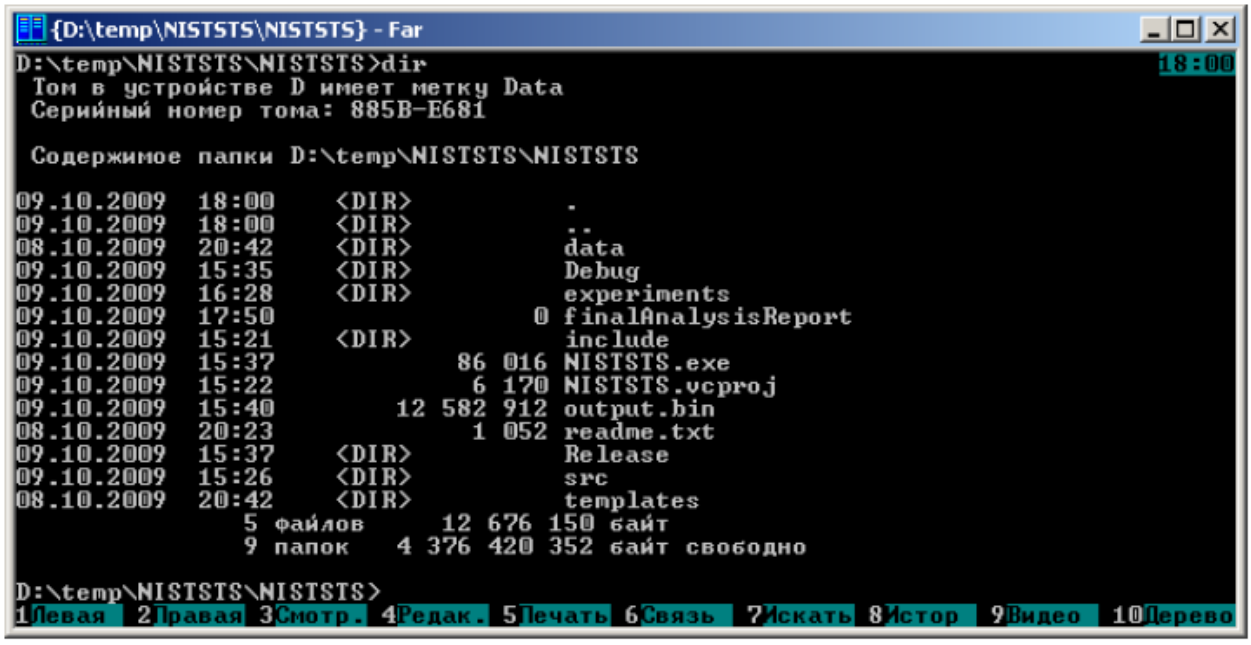


Рисунок 9: Информация о результатах тестирования

Приложение Б. Описание тестов NIST STS

1. Частотный побитовый тест

Суть данного теста заключается в определении соотношения между нулями и единицами во всей двоичной последовательности. Цель — выяснить, действительно ли число нулей и единиц в последовательности приблизительно одинаковы, как это можно было бы предположить в случае истинно случайной бинарной последовательности. Тест оценивает, насколько близка доля единиц к 0,5. Таким образом, число нулей и единиц должно быть примерно одинаковым. Если вычисленное в ходе теста значение вероятности $p <$ 0,01, то данная двоичная последовательность не является истинно случайной. В противном случае, последовательность носит случайный характер. Стоит отметить, что все последующие тесты проводятся при условии, что пройден данный тест.

2. Частотный блочный тест

Суть теста — определение доли единиц внутри блока длиной m бит. Цель — выяснить действительно ли частота повторения единиц в блоке длиной m бит приблизительно равна $m/2$, как можно было бы предположить в случае абсолютно случайной последовательности. Вычисленное в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p <$ 0,01), двоичная последовательность не носит истинно случайный характер. Если принять $m = 1$, данный тест переходит в тест № 1 (частотный побитовый тест).

3. Тест на последовательность одинаковых битов

Суть состоит в подсчёте полного числа рядов в исходной последовательности, где под словом «ряд» подразумевается непрерывная подпоследовательность одинаковых битов. Ряд длиной k бит состоит из k абсолютно идентичных битов, начинается и заканчивается с бита, содержащего противоположное значение. Цель данного теста — сделать вывод о том, действительно ли количество рядов, состоящих из единиц и нулей с различными длинами, соответствует их количеству в случайной последовательности. В частности, определяется быстро либо медленно чередуются единицы и нули в исходной последовательности. Если вычисленное в ходе теста значение вероятности $p <$ 0,01, то

данная двоичная последовательность не является истинно случайной. В противном случае, она носит случайный характер.

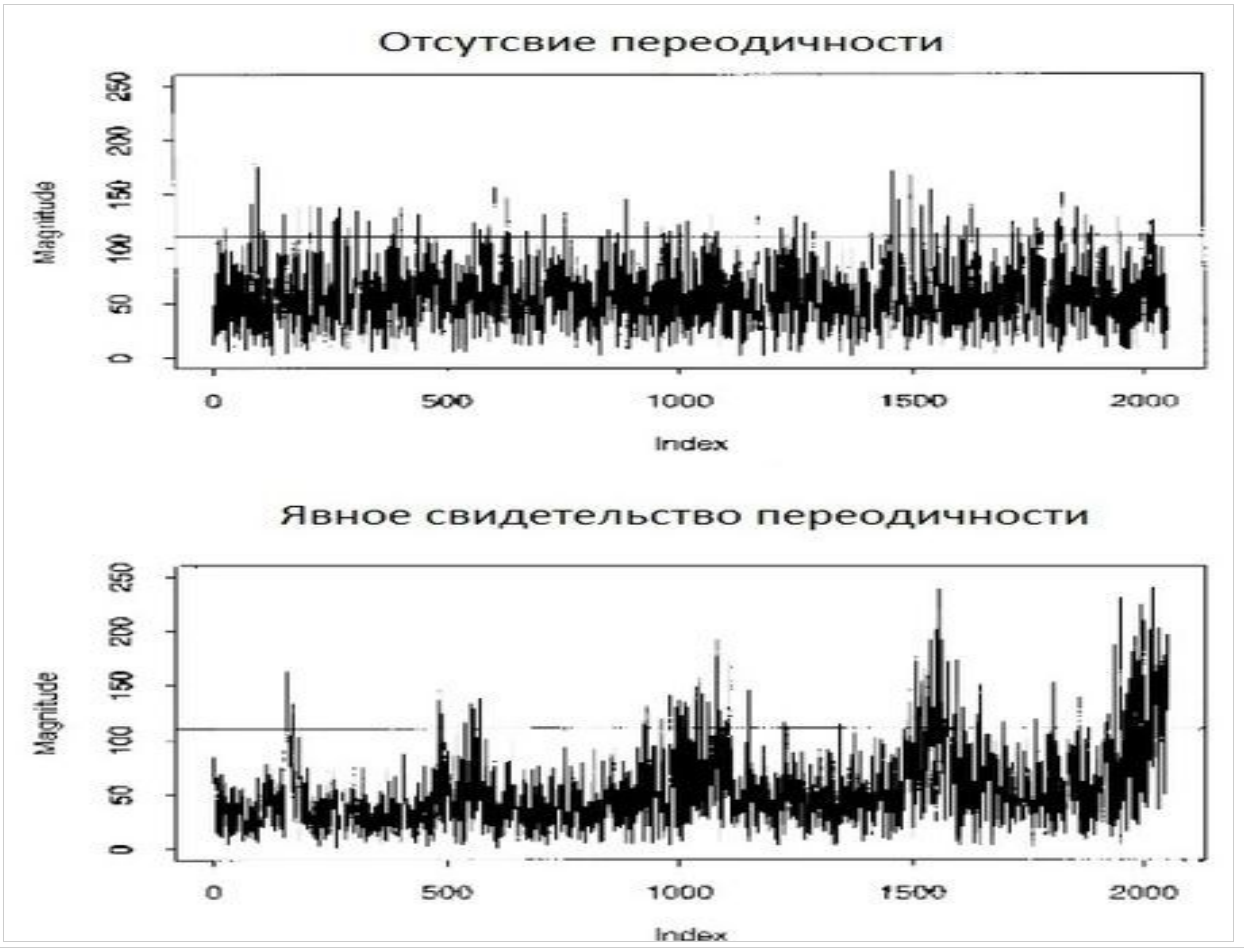
4. Тест на самую длинную последовательность единиц в блоке

В данном тесте определяется самый длинный ряд единиц внутри блока длиной m бит. Цель — выяснить действительно ли длина такого ряда соответствует ожиданиям длины самого протяжённого ряда единиц в случае абсолютно случайной последовательности. Если высчитанное в ходе теста значение вероятности $p < 0{,}01$ полагается, что исходная последовательность не является случайной. В противном случае, делается вывод о ее случайности. Следует заметить, что из предположения о примерно одинаковой частоте появления единиц и нулей (тест № 1) следует, что точно такие же результаты данного теста будут получены при рассмотрении самого длинного ряда нулей. Поэтому измерения можно проводить только с единицами.

5. Тест рангов бинарных матриц

Здесь производится расчёт рангов непересекающихся подматриц, построенных из исходной двоичной последовательности. Целью этого теста является проверка на линейную зависимость подстрок фиксированной длины, составляющих первоначальную последовательность. В случае, если вычисленное в ходе теста значение вероятности $p < 0{,}01$, делается вывод о неслучайном характере входной последовательности бит. В противном случае, считаем ее абсолютно случайной.

6. Спектральный тест



Суть теста заключается в оценке высоты пиков дискретного преобразования Фурье исходной последовательности. Цель — выявление периодических свойств входной последовательности, например, близко расположенных друг к другу повторяющихся

участков. Тем самым это явно демонстрирует отклонения от случайного характера исследуемой последовательности. Идея состоит в том, чтобы число пиков, превышающих пороговое значение в 95 % по амплитуде, было значительно больше 5 %. Если вычисленное в ходе теста значение вероятности $p < 0{,}01$, то данная двоичная последовательность не является абсолютно случайной. В противном случае, она носит случайный характер.

7. Тест на совпадение неперекрывающихся шаблонов

В данном тесте подсчитывается количество заранее определенных шаблонов, найденных в исходной последовательности. Цель — выявить генераторы случайных или псевдослучайных чисел, формирующие слишком часто заданные непериодические шаблоны. Как и в тесте № 8 на совпадение перекрывающихся шаблонов для поиска конкретных шаблонов длиной m бит используется окно также длиной m бит. Если шаблон не обнаружен, окно смещается на один бит. Если же шаблон найден, окно перемещается на бит, следующий за найденным шаблоном, и поиск продолжается дальше. Вычисленное в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p < 0{,}01$), двоичная последовательность не является абсолютно случайной.

8. Тест на совпадение перекрывающихся шаблонов

Суть данного теста заключается в подсчете количества заранее определенных шаблонов, найденных в исходной последовательности. Как и в тесте № 7 на совпадение неперекрывающихся шаблонов для поиска конкретных шаблонов длиной m бит используется окно также длиной m бит. Сам поиск производится аналогичным образом. Если шаблон не обнаружен, окно смещается на один бит. Разница между этим тестом и тестом № 7 заключается лишь в том, что если шаблон найден, окно перемещается только на бит вперед, после чего поиск продолжается дальше. Вычисленное в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p < 0{,}01$), двоичная последовательность не является абсолютно случайной.

9. Универсальный статистический тест Маурера

Здесь определяется число бит между одинаковыми шаблонами в исходной последовательности (мера, имеющая непосредственное отношение к длине сжатой последовательности). Цель теста — выяснить может ли данная последовательность быть значительно сжата без потерь информации. В случае, если это возможно сделать, то она не является истинно случайной. В ходе теста вычисляется значение вероятности p. Если $p < 0{,}01$, то полагается, что исходная последовательность не является случайной. В противном случае, делается вывод о её случайности.

10. Тест на линейную сложность

В основе теста лежит принцип работы линейного регистра сдвига с обратной связью (англ. *Linear Feedback Shift Register, LFSR*). Цель — выяснить является ли входная последовательность достаточно сложной для того, чтобы считаться абсолютно случайной. Абсолютно случайные последовательности характеризуются длинными линейными регистрами сдвига с обратной связью. Если же такой регистр слишком короткий, то предполагается, что последовательность не является в полной мере случайной. В ходе теста вычисляется значение вероятности p. Если $p < 0{,}01$, то полагается, что исходная последовательность не является случайной. В противном случае, делается вывод о её случайности.

11. Тест на периодичность

Данный тест заключается в подсчете частоты всех возможных перекрываний шаблонов длины m бит на протяжении исходной последовательности битов. Целью является определение действительно ли количество появлений $2m$ перекрывающихся шаблонов длиной m бит, приблизительно такое же как в случае абсолютно случайной

входной последовательности бит. Последняя, как известно, обладает однообразностью, то есть каждый шаблон длиной m бит появляется в последовательности с одинаковой вероятностью. Если вычисленное в ходе теста значение вероятности $p < 0{,}01$, то данная двоичная последовательность не является абсолютно случайной. В противном случае, она носит случайный характер. Стоит отметить, что при m=1 тест на периодичность переходит в частотный побитовый тест (№ 1).

12. Тест приблизительной энтропии

Как и в тесте на периодичность в данном тесте акцент делается на подсчёте частоты всех возможных перекрываний шаблонов длины m бит на протяжении исходной последовательности битов. Цель теста — сравнить частоты перекрывания двух последовательных блоков исходной последовательности с длинами m и $m+1$ с частотами перекрывания аналогичных блоков в абсолютно случайной последовательности. Вычисляемое в ходе теста значение вероятности p должно быть не меньше 0,01. В противном случае ($p < 0{,}01$), двоичная последовательность не является абсолютно случайной.

13. Тест кумулятивных сумм

Тест заключается в максимальном отклонении (от нуля) при произвольном обходе, определяемым кумулятивной суммой заданных (-1, +1) цифр в последовательности. Цель данного теста — определить является ли кумулятивная сумма частичных последовательностей, возникающих во входной последовательности, слишком большой или слишком маленькой по сравнению с ожидаемым поведением такой суммы для абсолютно случайной входной последовательности. Таким образом, кумулятивная сумма может рассматриваться как произвольный обход. Для случайной последовательности отклонения от произвольного обхода должны быть вблизи нуля. Для некоторых типов последовательностей, не являющихся в полной мере случайными подобные отклонения от нуля при произвольном обходе будут достаточно существенными. Если вычисленное в ходе теста значение вероятности $p < 0{,}01$, то входная двоичная последовательность не является абсолютно случайной. В противном случае, она носит случайный характер.

14. Тест на произвольные отклонения

Суть данного теста заключается в подсчёте числа циклов, имеющих строго k посещений при произвольном обходе кумулятивной суммы. Произвольный обход кумулятивной суммы начинается с частичных сумм после последовательности (0,1), переведённой в соответствующую последовательность (-1, +1). Цикл произвольного обхода состоит из серии шагов единичной длины, совершаемых в случайном порядке. Кроме того такой обход начинается и заканчивается на одном и том же элементе. Цель данного теста — определить отличается ли число посещений определенного состояния внутри цикла от аналогичного числа в случае абсолютно случайной входной последовательности. Фактически данный тест есть набор, состоящий из восьми тестов, проводимых для каждого из восьми состояний цикла: −4, −3, −2, −1 и +1, +2, +3, +4. В каждом таком тесте принимается решение о степени случайности исходной последовательности в соответствии со следующим правилом: если вычисленное в ходе теста значение вероятности $p < 0{,}01$, то входная двоичная последовательность не является абсолютно случайной. В противном случае, она носит случайный характер.

15. Другой тест на произвольные отклонения

В этом тесте подсчитывается общее число посещений определенного состояния при произвольном обходе кумулятивной суммы. Целью является определение отклонений от ожидаемого числа посещений различных состояний при произвольном обходе. В действительности этот тест состоит из 18 тестов, проводимых для каждого состояния: −9, −8, …, −1 и +1, +2, …, +9. На каждом этапе делается вывод о случайности входной

последовательности. Если вычисленное в ходе теста значение вероятности $p < 0{,}01$, то входная двоичная последовательность не является абсолютно случайной. В противном случае, она носит случайный характер.