

Лабораторная работа 3

Тема:

“Генерация псевдослучайных чисел”

Выполнил:

Студент группы 22207 Гордеев Никита

Задание:

- Программа, реализующую линейный конгруэнтный ГПСЧ при $n=24$, способную генерировать случайные числа как по одному (с выводом на экран), так и последовательностью произвольной длины (с записью в файл)).
- Значения, генерируемые ГПСЧ должны иметь равномерное распределение.
- Инициализирующая (генерация значений параметров ГПСЧ) и генерирующая части программы должны быть выполнены в виде отдельных функций (процедур).

Значения параметров разбиения:

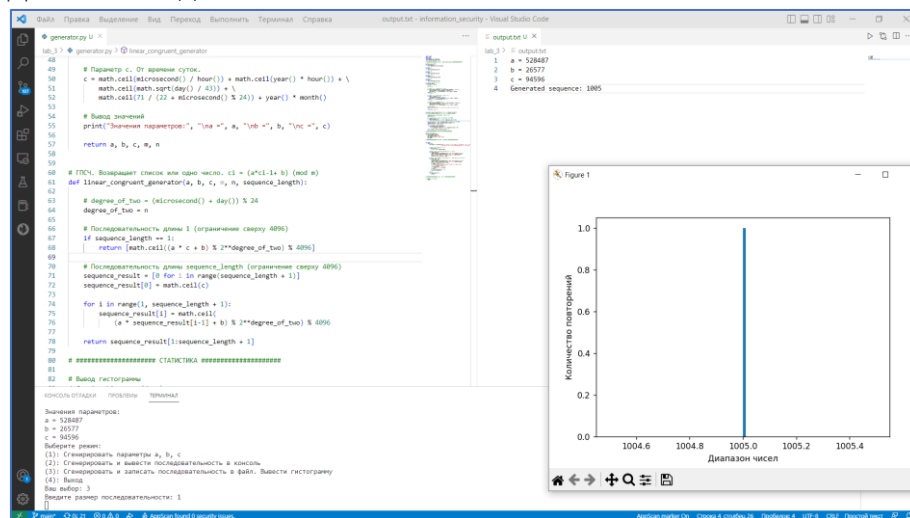
- Значения параметров a , b и c должны генерироваться случайным образом, но без использования стандартного генератора случайных чисел

Описание способа генерации значений параметров a , b и c

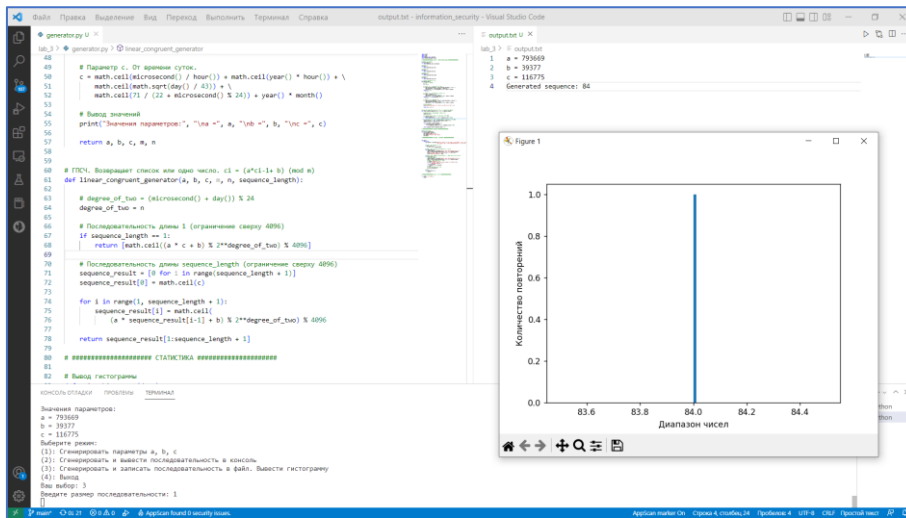
(и способ генерации случайных чисел, и действия в случае, когда сгенерированные числа не удовлетворяют требованиям к параметрам).

Статистика:

Длина последовательности = 1

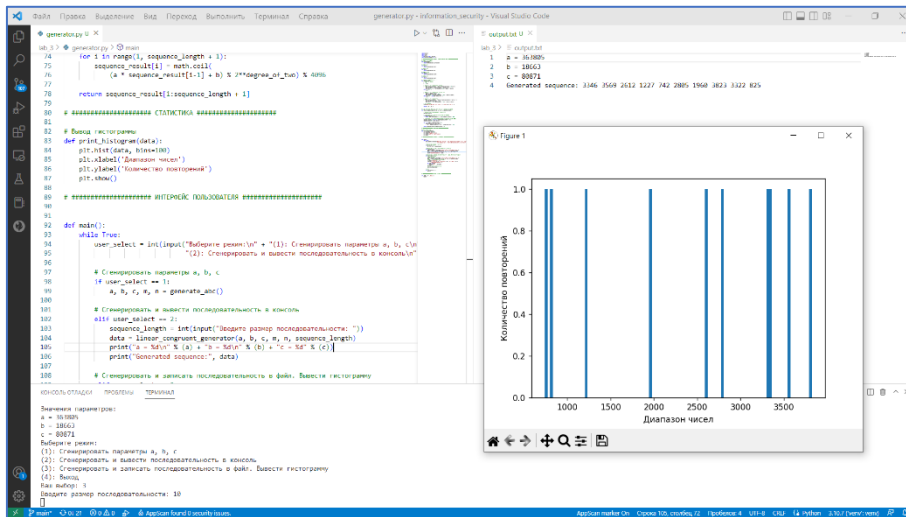


Тест 1

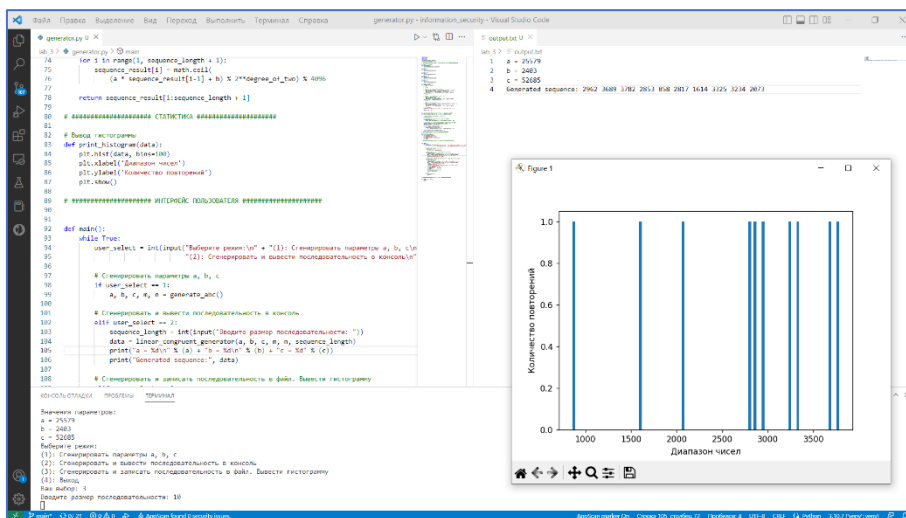


Тест 2

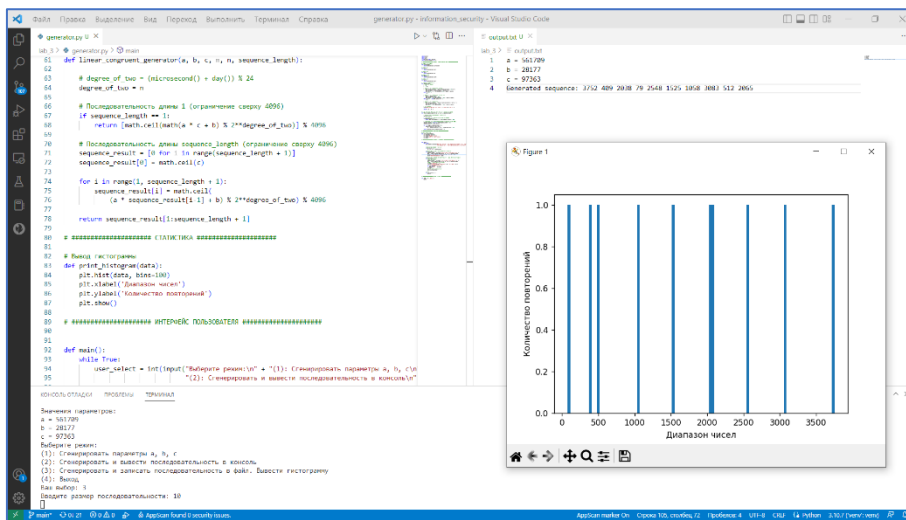
Длина последовательности = 10



Тест 3

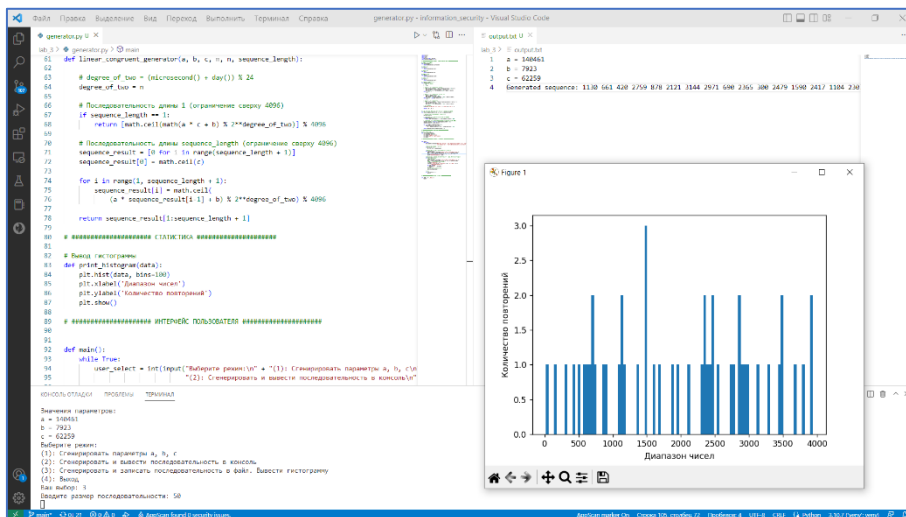


Тест 4

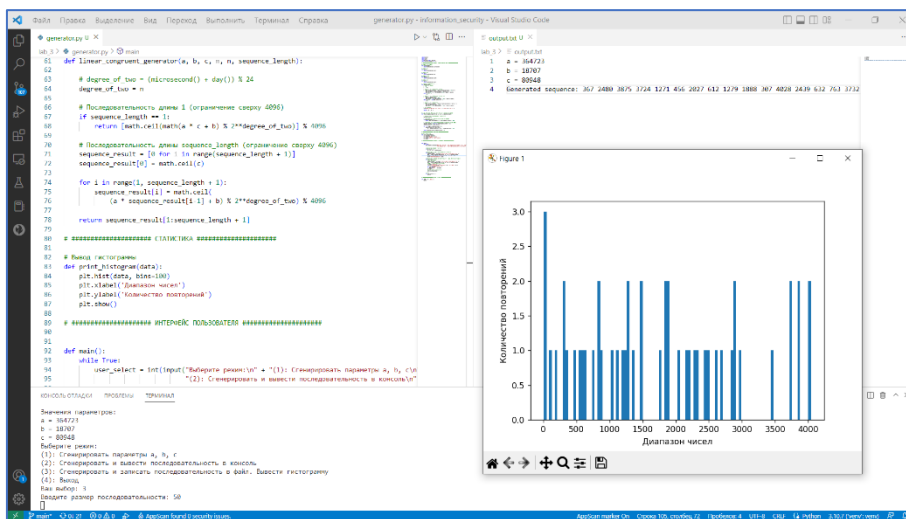


Тест 5

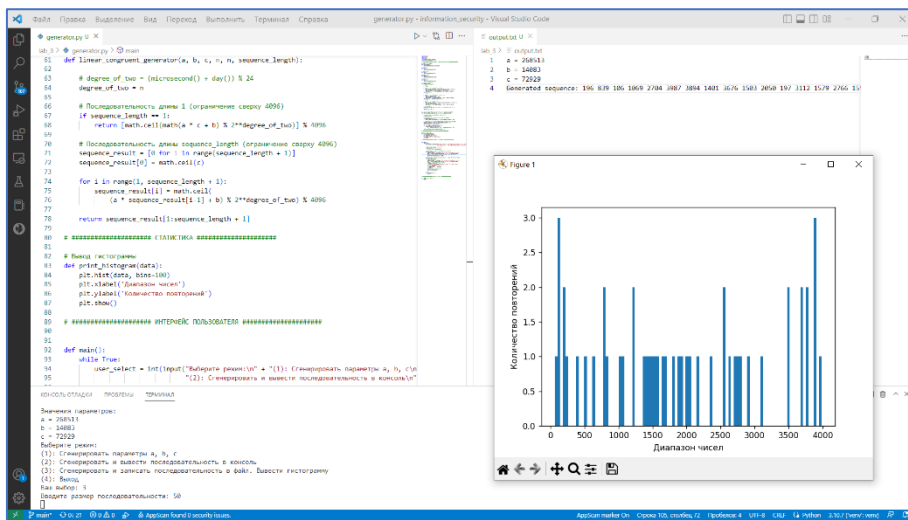
Длина последовательности = 50



Тест 6

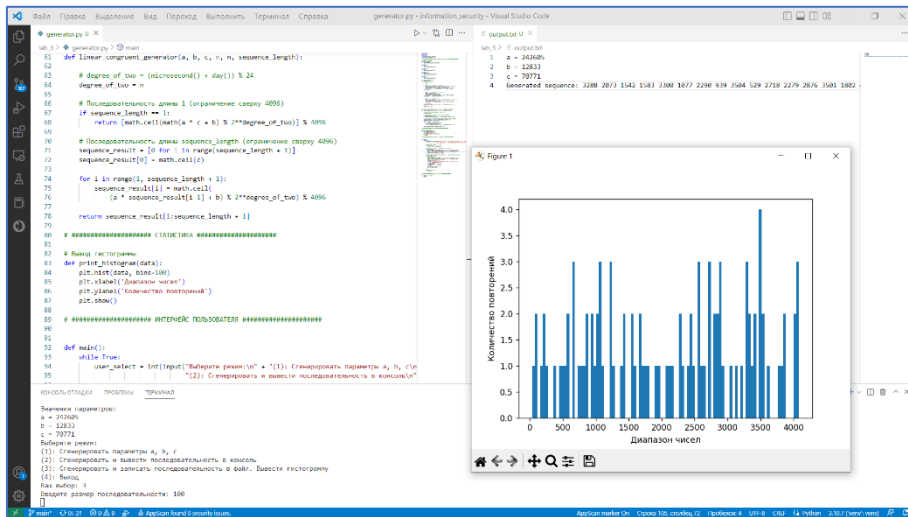


Тест 7

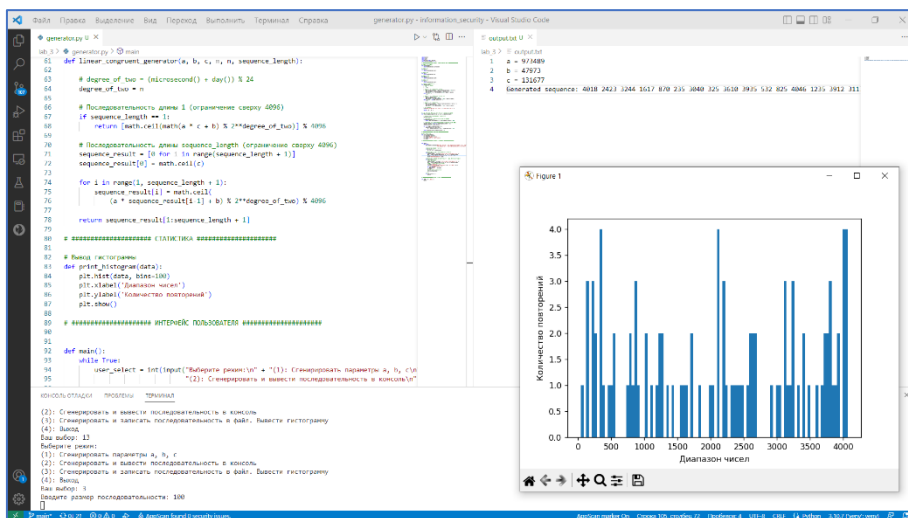


Тест 8

Длина последовательности = 100

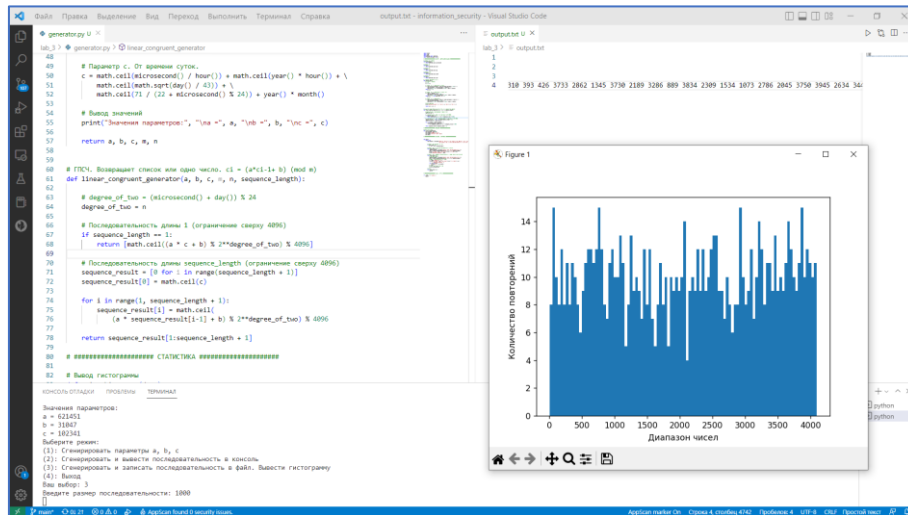


Тест 9



Тест 10

Длина последовательности = 1000



Тест 11

Текст функции (процедуры), выполняющей генерацию параметров a , b и c .

```
# Текущая микросекунда
def microsecond():
    return datetime.now().microsecond

# Текущий час
def hour():
    return datetime.now().hour

# Текущий день
def day():
    return datetime.now().day

# Текущий месяц
def month():
    return datetime.now().month

# Текущий год
def year():
    return datetime.now().year

# Получение случайных a и b и c
def generate_abc():
    n = 24
    m = 2 ** n

    # Параметр a. От времени суток, остаток.
    a = math.ceil(microsecond() / math.ceil(math.sqrt(day() / 57))) + \
        math.ceil(day() / math.ceil(year() * month())) + \
        math.ceil(34 / (7 + microsecond() % 24)) - year() * month()
    while a % 6 != 1:
        a += 1

    # Параметр b. От времени суток, остаток, НОД.
    b = year() * month() + 5 * \
        math.ceil(microsecond() / 104) + \
        math.ceil(31 / (12 + microsecond() % 14)) - year() * month()
    while b % 2 != 1 and math.gcd(b, m) != 1:
        b += 1

    # Параметр c. От времени суток.
    c = math.ceil(microsecond() / hour()) + math.ceil(year() * hour()) + \
        math.ceil(math.sqrt(day() / 43)) + \
        math.ceil(71 / (22 + microsecond() % 24)) + year() * month()

    # Вывод значений
    print("Значения параметров:", "\na =", a, "\nb =", b, "\nc =", c)

    return a, b, c, m, n
```

Текст функции (процедуры), реализующей линейный конгруэнтный ГПСЧ.

```
# ГПСЧ. Возвращает список или одно число. ci = (a*ci-1+ b) (mod m)
def linear_congruent_generator(a, b, c, m, n, sequence_length):

    # degree_of_two = (microsecond() + day()) % 24
    degree_of_two = n

    # Последовательность длины 1 (ограничение сверху 4096)
    if sequence_length == 1:
        return [math.ceil((a * c + b) % 2**degree_of_two) % 4096]

    # Последовательность длины sequence_length (ограничение сверху 4096)
    sequence_result = [0 for i in range(sequence_length + 1)]
    sequence_result[0] = math.ceil(c)

    for i in range(1, sequence_length + 1):
        sequence_result[i] = math.ceil(
            (a * sequence_result[i-1] + b) % 2**degree_of_two) % 4096

    return sequence_result[1:sequence_length + 1]
```

Пользовательский интерфейс.

```
# ##### СТАТИСТИКА #####

# Вывод гистограммы
def print_histogram(data):
    plt.hist(data, bins=100)
    plt.xlabel('Диапазон чисел')
    plt.ylabel('Количество повторений')
    plt.show()

# ##### ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ #####

def main():
    while True:
        user_select = int(input("Выберите режим:\n" + "(1): Сгенерировать параметры a, b, c\n" +
                                "(2): Сгенерировать и вывести последовательность в консоль\n" +
                                "(3): Сгенерировать и записать последовательность в файл. Вывести гистограмму\n" +
                                "(4): Выход\n" + "Ваш выбор: "))

        # Сгенерировать параметры a, b, c
        if user_select == 1:
            a, b, c, m, n = generate_abc()

        # Сгенерировать и вывести последовательность в консоль
        elif user_select == 2:
            sequence_length = int(input("Введите размер последовательности: "))
            data = linear_congruent_generator(a, b, c, m, n, sequence_length)
            print("a = %d\n" % (a) + "b = %d\n" % (b) + "c = %d\n" % (c))
            print("Generated sequence:", data)

        # Сгенерировать и записать последовательность в файл. Вывести гистограмму
        elif user_select == 3:
            # Генерация данных
            sequence_length = int(input("Введите размер последовательности: "))
            data = linear_congruent_generator(a, b, c, m, n, sequence_length)
            # Запись в последовательности в файл
            f = open('output.txt', 'w')
            f.write("a = %d\n" % (a) + "b = %d\n" % (b) + "c = %d\n" % (c) +
                    "Generated sequence: ")
            for item in data:
                f.write("%s " % str(item))
            f.close()
            # Печать гистограммы на экран
            print_histogram(data)

        # Выход
        elif user_select == 4:
            exit(0)
```

Материалы:

Random int without importing 'random' // stackoverflow URL:

<https://stackoverflow.com/questions/22950768/random-int-without-importing-random> (дата обращения: 07.12.2022).

Python Random Function without using random module // stackoverflow URL:

<https://stackoverflow.com/questions/28705965/python-random-function-without-using-random-module> (дата обращения: 07.12.2022).

Clock drift // Wikipedia URL: https://en.wikipedia.org/wiki/Clock_drift#Random_number_generators (дата обращения: 07.12.2022).

Datetime current year and month in Python // stackoverflow URL:

<https://stackoverflow.com/questions/28189442/datetime-current-year-and-month-in-python> (дата обращения: 07.12.2022).

Matplotlib.pyplot.hist() in Python // GeeksforGeeks URL: <https://www.geeksforgeeks.org/matplotlib-pyplot-hist-in-python/> (дата обращения: 07.12.2022).

Python math.fmod() Method // w3schools URL:

https://www.w3schools.com/python/ref_math_fmod.asp (дата обращения: 07.12.2022).

Math.ceil() // mdn web docs _ URL:

https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Math/ceil (дата обращения: 07.12.2022).

Writing printed output to a file (python) // stackoverflow URL:

<https://stackoverflow.com/questions/16190973/writing-printed-output-to-a-file-python> (дата обращения: 07.12.2022).