Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 3

Дисциплина: Моделирование

Тема: Генерация псевдослучайных последовательностей

Студент: Платонова О. С.

Группа: ИУ7-75Б

Оценка (баллы): _____

Преподаватель: Рудаков И. В.

Москва, 2021 г.

Цель работы: генерация псевдослучайных последовательностей 1/2/3 разрядных чисел алгоритмическим и табличным способом. Реализация критерия оценки случайности последовательности.

Случайные последовательности

Выделяют 3 способа получения последовательности случайных чисел:

1. аппаратный;
2. табличный/файловый;
3. алгоритмический.

1. Случайные числа вырабатываются генератором случайных чисел. Реализация не требует дополнительных вычислительных операций по выработке чисел.

2. Случайные числа записываются в файл (таблицу).
   Последовательность формируется на основе некоррелированных чисел, записанных в файл. Последовательность состоит из абсолютно случайных чисел. Однако для хранения большого количества цифр требуется много памяти. В данной реализации индекс элемента зависит от времени обращения.

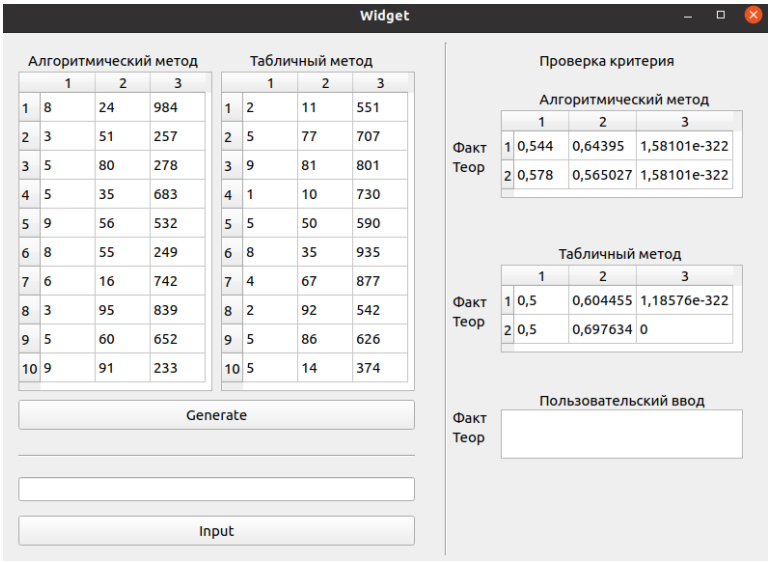
3. Последовательность случайных чисел периодична.
   В данной работе был реализован линейно-конгруэнтный метод. Реализованный алгоритм работает быстро и легок в реализации. Однако, последовательность состоит из псевдослучайных чисел и не обладает

большим периодом. В представленной реализации период не превышает 100000.

Критерий оценки случайности последовательности

Критерий является статистическим тестом и основан на вероятностных данных. Заключается в сравнении теоретического расчета и фактических результатов. Рассматривается интервал (MX – DX; MX + DX). Фактические данные описывают количество сгенерированных чисел, попавших в заданный интервал; а теоретический расчет описывает отношение длины указанного интервала к длине интервала, на котором генерируется последовательность.

Результаты

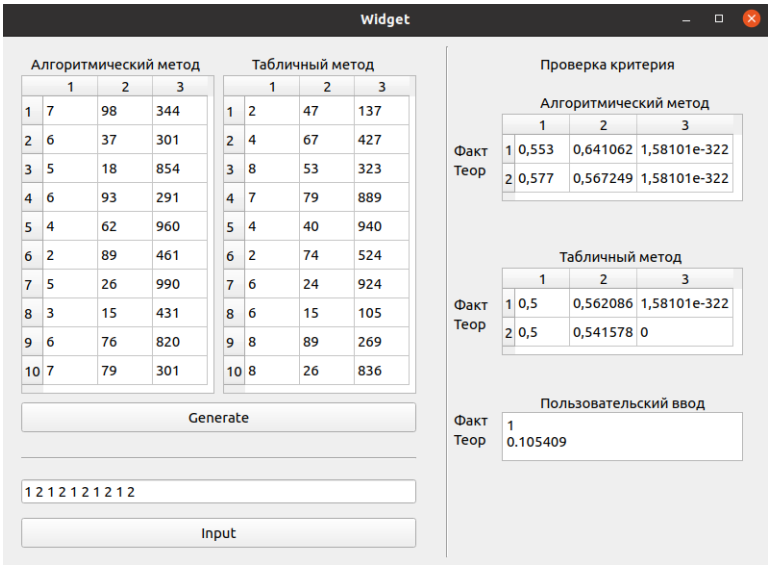


	1	2	3
1	8	24	984
2	3	51	257
3	5	80	278
4	5	35	683
5	9	56	532
6	8	55	249
7	6	16	742
8	3	95	839
9	5	60	652
10	9	91	233

	1	2	3
1	2	11	551
2	5	77	707
3	9	81	801
4	1	10	730
5	5	50	590
6	8	35	935
7	4	67	877
8	2	92	542
9	5	86	626
10	5	14	374

	1	2	3
1	0,544	0,64395	1,58101e-322
2	0,578	0,565027	1,58101e-322

	1	2	3
1	0,5	0,604455	1,18576e-322
2	0,5	0,697634	0



	1	2	3
1	7	98	344
2	6	37	301
3	5	18	854
4	6	93	291
5	4	62	960
6	2	89	461
7	5	26	990
8	3	15	431
9	6	76	820
10	7	79	301

	1	2	3
1	2	47	137
2	4	67	427
3	8	53	323
4	7	79	889
5	4	40	940
6	2	74	524
7	6	24	924
8	6	15	105
9	8	89	269
10	8	26	836

		1	2	3
Факт	1	0,553	0,641062	1,58101e-322
Теор	2	0,577	0,567249	1,58101e-322

		1	2	3
Факт	1	0,5	0,562086	1,58101e-322
Теор	2	0,5	0,541578	0

Листинг

Файл generations.cpp

```cpp
#include "generation.h"

#include <iostream>
#include <fstream>
#include <sstream>
#include <cmath>

using namespace std;

const string tableName = "/home/platosha/Desktop/BMSTU/7sem/Modeling-2/lab3/tableRandom.txt";
const int A = 73129;
const int C = 95121;
const int m = 100000;

vector<int> algGenerator(const int amount,
                         const int leftBorder, const int rightBorder)
{
    time_t now = time(0);
    tm *gmtm = gmtime(&now);
    static unsigned int seed = gmtm->tm_sec;

    vector<int> res;
    for (int i = 0; i < amount; i++) {
        seed = (seed * A + C) % m;
        int number = leftBorder + seed % (rightBorder - leftBorder + 1);
        res.push_back(number);
    }

    return res;
}
```

```cpp
vector<int> getTable()
{
    string line = "", number = "";
    vector<int> table;

    ifstream in(tableName);
    if (in.is_open()) {
        while (getline(in, line)) {
            stringstream strStream(line);
            while (getline(strStream, number, ' ')) {
                table.push_back(atoi(number.c_str()));
            }
        }
    }
    in.close();

    return table;
}

vector<int> tabGenerator(const int amount,
                         const int leftBorder, const int rightBorder)
{
    time_t now = time(0);
    tm *gmtm = gmtime(&now);
    unsigned int sec = gmtm->tm_sec;

    vector<int> table = getTable();

    vector<int> res;
    for (int i = 0; i < amount; i++) {
        int k = sec % 50, l = (sec + i) % 10;
        int number = leftBorder + table[10 * k + l] % (rightBorder - leftBorder + 1);
        res.push_back(number);
    }

    return res;
```

```cpp
vector<double> frequencyTest(const std::vector<int> sequence,
                             const int leftBorder, const int rightBorder)
{
    double avg = 0;
    for (size_t i = 0; i < sequence.size(); i++) {
        avg += sequence[i];
    }
    avg /= sequence.size();

    double disp = 0;
    for (size_t i = 0; i < sequence.size(); i++) {
        disp += (sequence[i] - avg) * (sequence[i] - avg);
    }
    disp /= (sequence.size() - 1);
    disp = sqrt(disp);

    int count = 0;
    for (size_t i = 0; i < sequence.size(); i++) {
        if ((avg - disp) < sequence[i] && sequence[i] < (avg + disp)) {
            count++;
        }
    }

    double resActual = static_cast<float> (count) / static_cast<float> (sequence.size());
    double resTheory = 2 * disp / (rightBorder - leftBorder);
    vector<double> res {resActual, resTheory};

    return res;
}
```

Вывод

На основе приведенных результатов можно сделать вывод о том, что согласно выбранному критерию, табличный метод лучше алгоритмического, который выполняет генерацию псевдослучайной последовательности с периодом не превосходящим 100000. Результаты критерия, полученные на основе последовательности с периодом 2 (ручной ввод) являются не удовлетворительными, т.к. разница между фактическими и теоретическими данными составляет 90% (> 50%).