|  |
| --- |
| Пензенский государственный университет  Факультет вычислительной техники  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №3  по дисциплине «Методы защиты компьютерных систем»  Вариант № 1 |
|  |
|  |
| Выполнили: студенты группы 19ВВ3:  Земляков В. Д.  Ерёмин А. А.  Проверил:  Дубравин А. В. |
| Пенза  2022 |

**Тема:** генераторы псевдослучайных последовательностей.

**Общие сведения:**

Генератор псевдослучайных чисел (ГПСЧ, англ. Pseudorandom number generator, PRNG) — алгоритм, генерирующий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному). Современная информатика широко использует псевдослучайные числа в самых разных приложениях — от метода Монте-Карло и имитационного моделирования до криптографии. При этом от качества используемых ГПСЧ напрямую зависит качество получаемых результатов.

**Задание**

Разработать генератор случайных чисел, используя метод в соответствии с номером варианта. Исследовать качество генератора, путем оценки распределения генерируемых чисел. Для этого необходимо разделить весь диапазон генерируемых чисел на 10 равных интервалов и подсчитать количество чисел, попадающих в каждый интервал. В качестве начального значения следует выбирать текущее время в формате Unix time (можно получить, используя функцию time\_t time(time\_t\* timer)).

**Метод генерации случайных чисел**

Метод серединных квадратов. Разрядность при вычислении 16 бит, разрядность результата 8 бит.

**Ход работы**

Листинг программы

#include <iostream>

#include <time.h>

#include <vector>

#include <algorithm>

#include <Windows.h>

class MidSquareRandom

{

private:

time\_t seed;

public:

MidSquareRandom()

{

this->seed = time(NULL);

}

MidSquareRandom(time\_t seed)

{

this->seed = seed;

}

uint8\_t Rand()

{

uint8\_t num = (uint8\_t)this->seed;

uint16\_t sqr;

uint8\_t res;

sqr = num \* num;

res = (sqr >> 4) & 0xFF;

this->seed = (time\_t)res;

return res;

}

void ChangeSeed(time\_t seed)

{

this->seed = seed;

}

};

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

MidSquareRandom rand = MidSquareRandom(1000);

int count\_random[10] = { 0 };

int num = 1;

std::vector<int> v;

while (num != 0)

{

num = rand.Rand();

printf("%d\n", num);

}

printf("\n");

printf("Распределение чисел: ");

for (int i = 1001; i < 10000; i++)

{

while (1)

{

if (num == 0)

{

count\_random[0]++;

break;

}

if (std::find(v.begin(), v.end(), num) != v.end())

{

break;

}

v.push\_back(num);

if (num < 25)

{

count\_random[0]++;

}

if (num > 25 && num < 50)

{

count\_random[1]++;

}

if (num > 50 && num < 75)

{

count\_random[2]++;

}

if (num > 75 && num < 100)

{

count\_random[3]++;

}

if (num > 100 && num < 125)

{

count\_random[4]++;

}

if (num > 125 && num < 150)

{

count\_random[5]++;

}

if (num > 150 && num < 175)

{

count\_random[6]++;

}

if (num > 175 && num < 200)

{

count\_random[7]++;

}

if (num > 200 && num < 225)

{

count\_random[8]++;

}

if (num > 225 && num < 256)

{

count\_random[9]++;

}

num = rand.Rand();

}

rand.ChangeSeed(i);

num = rand.Rand();

v.clear();

}

for (int i = 0; i < 10; i++)

{

std::cout << count\_random[i] << " ";

}

printf("\n");

}

**Результат работы программы**

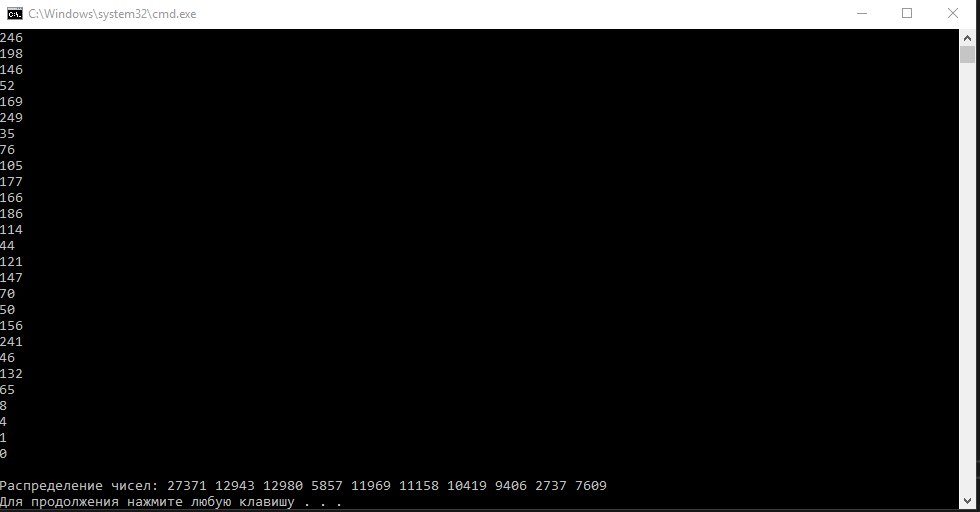


рисунок 1. Результат работы ГПСЧ с зерном 1000, а также распределение чисел в диапазоне зерна от 1001 до 10000.

**Вывод**

В ходе работы познакомились с алгоритмами ГПСЧ. Разработали программу генерации псевдослучайных последовательностей по алгоритму в соответствии с вариантом. Узнали об особенностях метода серединных квадратов. Недостатками данного метода являются: существование «циклов», когда последовательность начинает повторяться и уход алгоритма в 0.