Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 3

по курсу «Защита информации в сети Internet»

«Генераторы псевдослучайных последовательностей»

Выполнили студенты группы 18ВВ1:  
Гаевский И.

Воеков И.

Немцев А.

Приняли:  
к.т.н., доцент Дубравин А.В.

к.т.н., доцент Карамышева Н.С.

Пенза 2020

### Цель работы

### Разработать генератор псевдослучайных чисел.

### Лабораторное 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)).

|  |  |
| --- | --- |
| Номер варианта | Метод генерации случайных чисел |
| 3. Метод перемешивания. | Разрядность при вычислении 16 бит, разрядность результата 8 бит. |

**Метод перемешивания**

В методе перемешивания используются операции циклического сдвига содержимого ячейки влево и вправо. Идея метода состоит в следующем. Пусть в ячейке хранится начальное число R0. Циклически сдвигая содержимое ячейки влево на 1/4 длины ячейки, получаем новое число R0\*. Точно так же, циклически сдвигая содержимое ячейки R0 вправо на 1/4 длины ячейки, получаем второе число R0\*\*. Сумма чисел R0\* и R0\*\* дает новое случайное число R1. Далее R1 заносится в R0, и вся последовательность операций повторяется (см. рис.).

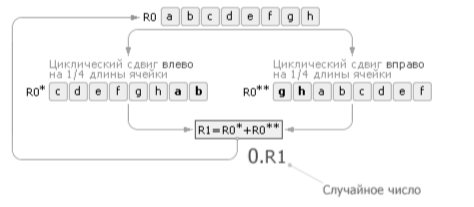


Рис. Схема метода серединных произведений

**Листинг**

#include <Windows.h>

#include <time.h>

#include <stdio.h>

#define SIZE\_MASS 10

#define COUNT\_ITER 1000

// Оценка распределения генерируемых чисел

void distribution(UINT8 integer, UINT8 freq[SIZE\_MASS])

{

double temp = static\_cast<double>(integer);

temp = temp / 255;

if (temp <= 0.1)

freq[0]++;

else if (temp <= 0.2)

freq[1]++;

else if (temp <= 0.3)

freq[2]++;

else if (temp <= 0.4)

freq[3]++;

else if (temp <= 0.5)

freq[4]++;

else if (temp <= 0.6)

freq[5]++;

else if (temp <= 0.7)

freq[6]++;

else if (temp <= 0.8)

freq[7]++;

else if (temp <= 0.9)

freq[8]++;

else if (temp <= 1)

freq[9]++;

}

int main()

{

UINT8 freq[SIZE\_MASS] = { 0 };

UINT16 startnum = (UINT16)time(0); // Начальное значение

UINT16 sum = startnum;

for (int i = 0; i < COUNT\_ITER; i++)

{

UINT16 num1 = (sum >> 4) | (sum << (16 – 4)); // Сдвиг вправо на четверть длины ячейки

UINT16 num2 = (sum << 4) | (sum >> (16 – 4)); // Сдвиг влево на четверть длины ячейки

sum = num1 + num2; // Сумма двух предыдущих чисел дает новое случайное число

UINT8 res = static\_cast<UINT8>(sum); // Результат должен быть 8-битным

printf("%d\n", res); // Выводим на экран псевдослучайное число

distribution(res, freq); // Оценка распределения генерируемых чисел

}

// Выводим на экран оценку распределения генерируемых чисел

printf("\nDistribution for %d:\n", startnum);

for (int i = 0; i < SIZE\_MASS; i++)

{

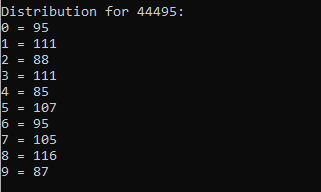
printf("%d = %d\n", i , freq[i]);

}

return 0;

}

**Результат работы программы**



Риc. Результат работы программы при начальном значении, заданным текущим временем

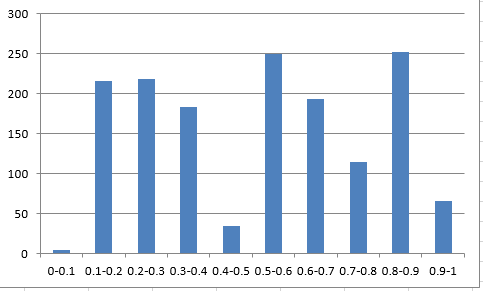


Рис. Оценка распределения генерируемых чисел при начальном значении, заданным текущим временем

**Вывод**

Разработали генератор псевдослучайных чисел с помощью метода перемешивания.