Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ**

отчет о лабораторной работе № 1

по дисциплине

*ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ*

***ВАРИАНТ 13***

|  |  |  |
| --- | --- | --- |
| Выполнила: | ст. гр. 230711 | Павлова В.С. |
| Проверил: | асс. каф. ИБ | Курбаков М.Ю. |

Тула, 2022 г.

**ЦЕЛЬ И ЗАДАЧА РАБОТЫ**

**Цель:** изучить способы генерации случайных чисел.

**Задача:** в данной работе требуется написать программу, демонстрирующую использование изученных принципов.

**ЗАДАНИЕ НА РАБОТУ**

Выполнить задание варианта 11 с использованием метода Р. Ковэю.

**Задание из варианта 11:** написать программу, определяющую, какое число чаще других встречается в последовательности целых случайных чисел. Для генерации чисел использовать линейный конгруэнтный метод.

# **СХЕМА ПРОГРАММЫ**

Схема алгоритма программы, определяющей, какое число чаще других встречается в последовательности целых случайных чисел, представлена на рисунках 1-4.



Рисунок 1 – Схема алгоритма программы определяющей, какое число чаще других встречается в последовательности целых случайных чисел



Рисунок 1 – Схема алгоритма программы определяющей, какое число чаще других встречается в последовательности целых случайных чисел (продолжение)



Рисунок 2 – Схема алгоритма подпрограммы для генерации целых случайных чисел по методу Р. Ковэю

|  |  |
| --- | --- |
| Рисунок 3 – Схема алгоритма подпрограммы для вывода массива | Рисунок 4 – Схема алгоритма подпрограммы для инициализации массива нулями |

# **ТЕКСТ ПРОГРАММЫ**

Текст программы на языке программирования С++, определяющей, какое число чаще других встречается в последовательности целых случайных чисел, представлен в листинге 1.

## **Листинг 1. Текст программы**

#include <iostream>

#include <cmath>

#include <iomanip>

#include <algorithm>

#include <vector>

using namespace std;

int RandomKoveu(int y, int m)

{

int x = y \* (y + 1) % m;

return x;

}

void PrintArray(long int\* a, int size)

{

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

{

cout << setw(3) << a[i] << " " << setw(5);

if (i != 0 && i % 10 == 0) cout << "\n";

}

cout << "\n";

return;

}

void InitializeByZero(int\* a, int size)

{

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

{

a[i] = 0;

}

return;

}

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

long int n, x0 = 10, m = 2, degree;

cout << "Введите длину последовательности: ";

cin >> n;

long int\* randomNumbersArray = new long int[n];

randomNumbersArray[0] = x0 \* 4 + 2;

cout << "Введите e (2<e<15) - степень двойки для модуля: ";

cin >> degree;

## **Листинг 1. Текст программы (продолжение)**

m = (int)pow(2, degree);

for (int i = 1; i < n; i++)

{

randomNumbersArray[i] = RandomKoveu(randomNumbersArray[i - 1], m);

}

PrintArray(randomNumbersArray, n);

long long int commonValuesSize = \*max\_element(randomNumbersArray, randomNumbersArray + n);

int\* commonValues = new int[commonValuesSize];

InitializeByZero(commonValues, commonValuesSize);

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

{

commonValues[randomNumbersArray[i]]++;

}

auto maxValue = max\_element(commonValues, commonValues + commonValuesSize);

cout << "\nСамый частый элемент в последовательности: "

<< distance(commonValues, maxValue)

<< ", он повторился " << commonValues[distance(commonValues, maxValue)] << " раз.";

return 0;

}

# **ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ**

Данная программа предназначена для нахождения самого частого числа в последовательности целых чисел. Последовательность генерируется квадратичным методом Р. Ковэю. При запуске программы пользователю предлагается ввести длину последовательности, а также степень двойки для модуля. На вводимые данные накладываются следующие ограничения:

* n – длина последовательности – целое число,
* degree (e) – степень двойки модуля –

После генерации последовательности по методу Р. Ковэю и обработки полученных данных программа выводит ответ – самое частое число и частота его появления. Если таких чисел несколько, выводится первое найденное.

**ИНСТРУКЦИЯ ПРОГРАММИСТА**

Данная программа предназначена для генерации последовательности квадратичным методом Р. Ковэю и нахождения самого частого числа в массиве. Структуры данных, используемые в программе, приведены в таблице 1.

Таблица 1 – Структуры данных в программе

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| n | Long int | Длина последовательности |
| degree | Long int | Степень двойки модуля |
| x0 | Long int | Начальное значение последовательности |
| randomNumbersArray | Long int\* | Массив для хранения последовательности |
| commonValues | Long long int\* | Массив для подсчёта повторений по индексам |

Программа разбита на 3подпрограммы:

1) int RandomKoveu(int y, int m) – подпрограмма для генерации случайных чисел методом Ковэю. Структуры данных, используемые в подпрограмме, приведены в таблице 2.

Таблица 2 – Структуры данных, используемые в подпрограмме RandomKoveu

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| *формальные параметры* | | |
| y, m | int | Предыдущее число в последовательности и модуль |

2) void PrintArray(long int\* a, int size) – подпрограмма для вывода массива. Структуры данных, используемые в подпрограмме, приведены в таблице 3.

Таблица 3 – Структуры данных, используемые в подпрограмме PrintArray

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| *формальные параметры* | | |
| size | int | Размер массива |
| a | long int\* a | Ссылка на начало выводимого массива |

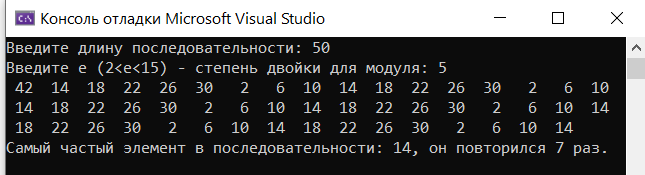
3) void InitializeByZero(int\* a, int size) – подпрограмма для заполнения массива нулями. Структуры данных, используемые в подпрограмме, приведены в таблице 4.

Таблица 4 – Структуры данных, используемые в подпрограмме InitializeByZero

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Предназначение** |
| *формальные параметры* | | |
| size | int | Размер массива |
| a | int\* a | Ссылка на начало выводимого массива |

# **ДЕМОНСТРАЦИОННЫЙ ПРИМЕР**

Результат работы программы изображен на рисунке 5.

  
  
Рисунок 5 – Пример работы программы

Проверим результат работы программы аналитически. Известно, что период при использовании квадратичного метода Ковэю U = = N/4. Исходя из введенных данных, максимальный период будет равен = 8, то есть каждые 8 чисел происходит повторение, тогда несложно подсчитать, что для 50 чисел период повторится 6 раз и начнется заново с числа 14, которое и окажется самым частым, то есть повторится уже 7 раз.

Если сравнить полученный результат, можно убедиться, что программа работает верно.

# **ВЫВОДЫ**

Руководствуясь полученными в ходе выполнения лабораторной работы сведениями, можно сделать вывод о том, что случайные числа, генерируемые в компьютере не являются такими уж и случайными. Генератор псевднослучайных чисел позволяет получать последовательности независимых чисел с определенным законом распределения, которые широко используются в компьютерной технике, начиная криптографией и заканчивая производством игр.

Современные программные генераторы, как правило, используют сложные математические алгоритмы. В данной лабораторной работе был рассмотрен такой алгоритм генерации равномерно распределенных случайных чисел, как квадратичный метод Р. Ковэю, который, благодаря использованию степеней двоек позволяет получать достаточно большие периоды, а благодаря возможности выбирать степень двойки, равную числу двоичных разрядов, нужных для записи целого числа в ЭВМ, можно добиться ещё и повышения компьютерного быстродействия при такой генерации.