Лабораторная работа № 3

Генерация псевдослучайных чисел

Во многих случаях, например, при создании ключей шифрования, требуется получить некоторое случайное число, которое и будет ключом. Для получения последовательности случайных (точнее, псевдослучайных, то есть периодически повторяющихся) чисел используются генераторы псевдослучайных чисел (ГПСЧ). В качестве примера простейшего ГПСЧ можно привести линейный конгруэнтный генератор, использующий рекуррентное соотношение вида:

$$\mathbf{c}_i = (\mathbf{a}*\mathbf{c}_{i-1} + \mathbf{b}) \pmod{\mathbf{m}}$$

где

$\mathbf{c}_i$ − i-е число псевдослучайной последовательности;

$\mathbf{c}_{i-1}$ − предшествующее число последовательности;

$\mathbf{c}_0$ − порождающее число последовательности;

a,**b**,**m** – натуральные числа (константы), удовлетворяющие определенным требованиям.

Значение **m** берется равным 2^n (где **n** – целое число). Период повторения псевдослучайных чисел зависит от выбираемых значений всех параметров, но никогда не превышает значения **m**.

Значение **b** должно быть нечетным числом, взаимно простым с **m**, а коэффициент **a** должен быть таким, чтобы **a** mod 4=1 (mod – остаток при делении с остатком). Значение $\mathbf{c}_0$ выбирается случайным образом. Значения **a**, **b** и $\mathbf{c}_0$ должны быть менее **m**.

Задание

Составьте программу, реализующую линейный конгруэнтный ГПСЧ при **n**=24 (то есть **m=**2^{24}), и способную генерировать случайные числа как по одному (с выводом на экран), так и последовательностью произвольной длины (с записью в файл). Значения параметров **a**, **b** и $\mathbf{c_0}$ должны генерироваться случайным образом, но без использования стандартного генератора случайных чисел в выбранном вами языке программирования. Необходимо придумать и реализовать способ получения случайных чисел в зависимости от времени суток, текущего положения курсора мыши или иных придуманных вами параметров. Если сгенерированное таким образом значение параметра не удовлетворяет требованиям, оно может каким-либо способом дополнительно изменяться.

Значения, генерируемые ГПСЧ должны иметь равномерное распределение. Для проверки этого, весь диапазон возможных значений ГПСЧ [0, **m**–1] делится на 100 интервалов равной длины (с округлением длины до целого, длина последнего интервала может отличаться от длины остальных). Во время или после генерации чисел подсчитываются относительные частоты попадания сгенерированных значений в каждый из интервалов (отношение числа значений, попавших в интервал, к общему числу сгенерированных значений). Затем вычисляется среднее арифметическое от относительных частот (в идеальном случае это будет 0.01 или 1%, то есть в каждый из ста интервалов попадет одна сотая от всех сгенерированных значений). Относительные частоты округляются либо минимум до 4 знаков после запятой (0.0079) в случае работы с

числовыми значениями, либо минимум до двух в случае работы с процентными (0,79%).

Итого, от программы требуется:

1) По запросу пользователя генерировать новые параметры ГПСЧ.

2) На основании текущих параметров ГПСЧ генерировать и выводить на экран следующее псевдослучайное число (столько раз, сколько понадобится пользователю).

3) На основании текущих параметров ГПСЧ сгенерировать последовательность псевдослучайных чисел заданной пользователем длины, записать её в файл, а на экран вывести гистограмму относительных частот попаданий сгенерированных значений в сто интервалов. При этом первое число генерируемой последовательности всегда должно рассчитываться на основании порождающего числа последовательности $\mathbf{c_0}$.

Пояснение к реализации пунктов 2 и 3 – если пользователь просмотрит по одному первые 10 чисел псевдослучайной последовательности на экране (пункт 2), а затем сгенерирует последовательность из 10 чисел (пункт 3), то эти последовательности совпадут. Далее, если пользователь сгенерирует последовательность из 15 чисел (снова пункт 3), то первые десять в ней совпадут с прошлой последовательностью. А если после этого пользователь попросит вывести следующее число на экран (пункт 2), то увидит одиннадцатое число из второй последовательности. Разумеется, при смене параметров ГПСЧ, должна изменяться и генерируемая последовательность.

Инициализирующая (генерация значений параметров ГПСЧ) и генерирующая части программы должны быть выполнены в виде

отдельных функций (процедур).

Отчет о выполнении лабораторной работы должен включать:

1. Титульный лист.
2. Описание способа генерации значений параметров **a**, **b** и $\mathbf{c_0}$ (и способ генерации случайных чисел, и действия в случае, когда сгенерированные числа не удовлетворяют требованиям к параметрам).
3. ***СТАТИСТИКА:*** Результаты проверки равномерности распределения линейного конгруэнтного ГПСЧ – средние, максимальные и минимальные относительные частоты попаданий в интервал, с указанием тех наборов значений **a**, **b** и $\mathbf{c_0}$, с которыми выполнялась проверка, а также длинами генерируемых последовательностей).
4. Текст функции (процедуры), выполняющей генерацию параметров **a**, **b** и $\mathbf{c_0}$.
5. Текст функции (процедуры), реализующей линейный конгруэнтный ГПСЧ.

Программа должна обеспечивать удобный пользовательский интерфейс. Все вводимые и выводимые данные должны сопровождаться четкими и ясными для пользователя пояснениями.

Работоспособность программы проверяется преподавателем.

Самоконтроль

1. Задайте длину последовательности равную 10 элементам. Визуально проверьте адекватность построения гистограммы. В идеальном случае она будет содержать ровно 10 столбцов "единичной" высоты и 90 "нулевых" столбцов. Вариант с наличием одного интервала, в который

попали два элемента, также является теоретически возможным и допустимым. Наличие более чем одного интервала с двумя попаданиями или появление интервала, в который попали три и более элементов является признаком ошибочной реализации ГПСЧ.

2. В случае 100 элементов максимальное количество попаданий в один интервал равно трем. Большее количество попаданий снова свидетельствует об ошибках в реализации ГПСЧ.

3. **HINT:** Однократное и не повторяющееся при перезапусках программы (с другими значениями **a**, **b** и $\mathbf{c_0}$) наблюдение чрезмерно большого количества попаданий в какой-либо интервал вполне допустимо, так как оно является побочным эффектом несовершенства процедуры генерации **a**, **b** и $\mathbf{c_0}$. Если ситуация с "чрезмерным количеством попаданий" воспроизводится регулярно, следует изменить способ генерации **a**, **b** и $\mathbf{c_0}$ и повторить тестирование.