

RND: Случайные числа

Задача: написать программу, генерирующую случайные числа и строящую гистограмму их распределения. Для упрощения работы преподавателем может выдаваться **шаблонный файл**.

Пишем по шагам:

1. Заполнить массив десятью случайными числами из полуоткрытого интервала $[a, b)$. Для этого:

- Написать функцию `double Random(double a, double b)`, возвращающую псевдослучайное вещественное число $a \leq x < b$.

Для этого воспользоваться стандартным библиотечным генератором псевдослучайных чисел (ГПСЧ) — функцией `rand()`, которая возвращает **целые** СЧ от 0 до `RAND_MAX` включительно (библиотечная константа, потенциально разная на разных компиляторах).

По целым числам можно, путём нормализации (деления на `RAND_MAX + 1`), получить вещественное из интервала $[0, 1)$. Для приведения его к интервалу $[a, b)$ необходимо домножить число на длину интервала и прибавить левую границу — обычное линейное преобразование:

$$x = a + \frac{n}{N_{\max} + 1} \cdot (b - a)$$

- Написать функцию `void FillRandom(double arr[], int size, double a, double b)`, заполняющую массив числами, вызывая `Random()` в цикле.
 - Написать функцию `void Print(double const arr[], int size)`, красиво печатающую массив: на отдельной строке, в фигурных скобках, через запятую. Для теста ГСЧ заполнить и напечатать массив из 10 чисел в промежутке $[0.2, 0.6)$.
2. Узнать у пользователя количество чисел для эксперимента, заполнить массив случайными числами и построить горизонтально расположенную гистограмму. Проверить для 10, 100, 10000 чисел. Для этого:
- Написать функцию `void BuildHistogram(double const arr[], int size, double a, double b, int spans[], int numSpans)`, разбивающую интервал на `numSpans` частей и подсчитывающую в массиве `spans`, сколько элементов массива `arr` попало в соответствующий подынтервал. Номер подынтервала:

$$i = \left\lfloor \frac{x - a}{b - a} \cdot N \right\rfloor$$

- Написать функцию `void PrintHistogram(int spans[], int numSpans, int width, double scale)`, печатающую горизонтальную гистограмму, по одной строчке на интервал. Для печати одной строчки написать функцию `void PrintHistogramLine(double value, int width)`. Она должна часть линии

представить в виде значков #, а остальную дополнить точками до ширины width.

3. Написать функцию `double RandomSkewed(double a, double b)`, возвращающую **неравномерно** распределённое ПСЧ. Она позволит еще лучше проверить, как хорошо строится гистограмма. Можно взять линейно распределённую величину и получить треугольник на гистограмме. Для интервала $[0, 1)$ такую величину можно получить из равномерной при помощи корня, и перевести в $[a, b)$ как обычно:

$$y = a + (b - a)\sqrt{x}$$

На “пять с плюсом” написать ещё одну функцию, отображающую гистограмму вертикально, а не на боку. На “тройку” должен присутствовать как минимум правильно сделанный пункт 1 и какие-то попытки написать пункт 2. Без них — неуд.

Тест ГСЧ: { 0.583, 0.893, 0.329, 0.988, 0.427, 0.399, 0.497, 0.024, 0.882 }

Введите количество испытаний: 10000

Равномерная СВ:

```
.....
.....
.....
.....
.....
#####.....
#####.....
#####.....
#####.....
#####.....
```

Скошенная СВ:

```
.....
.....
.....
.....
.....
#.....
###.....
#####.....
#####.....
#####.....
```