

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ
компьютерной безопасности и
криптографии

ТЕОРИЯ ПСЕВДОСЛУЧАЙНЫХ ГЕНЕРАТОРОВ

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ КУРСУ

студента 4 курса 431 группы

факультета компьютерных наук и информационных технологий

Мухи Семена Андреевича

фамилия, имя, отчество

Научный руководитель

Ст. преподаватель
Слеповичев

И.И.

подпись, дата

Саратов 2024

Задание 2. Преобразование ПСЧ к заданному распределению

Необходимо создать программу, преобразовывающую ПСЧ в другую последовательность ПСЧ с заданным распределением. Входные параметры для алгоритмов передаются с помощью файла, название которого указывается в строке параметров запуска программы (p1, p2, p3). Преобразованные числа записываются в файл distr-xx.dat, где <xx> – код распределения. На вход в качестве последовательности ПСЧ подаётся последовательность, сгенерированная с помощью аддитивного генератора с входными параметрами: /g:add /n:10000 /i:128;9;49.

1. Стандартное равномерное распределение с заданным интервалом

Описание алгоритма:

Функция распределения $0 \leq x < 1: f(x) = 1$

Если стандартное равномерное случайное число U получено методом, установленным в предыдущем параграфе, то равномерное случайное число должно быть получено в соответствии со следующей формулой: $Y = b * U + a$.

Параметры запуска программы:

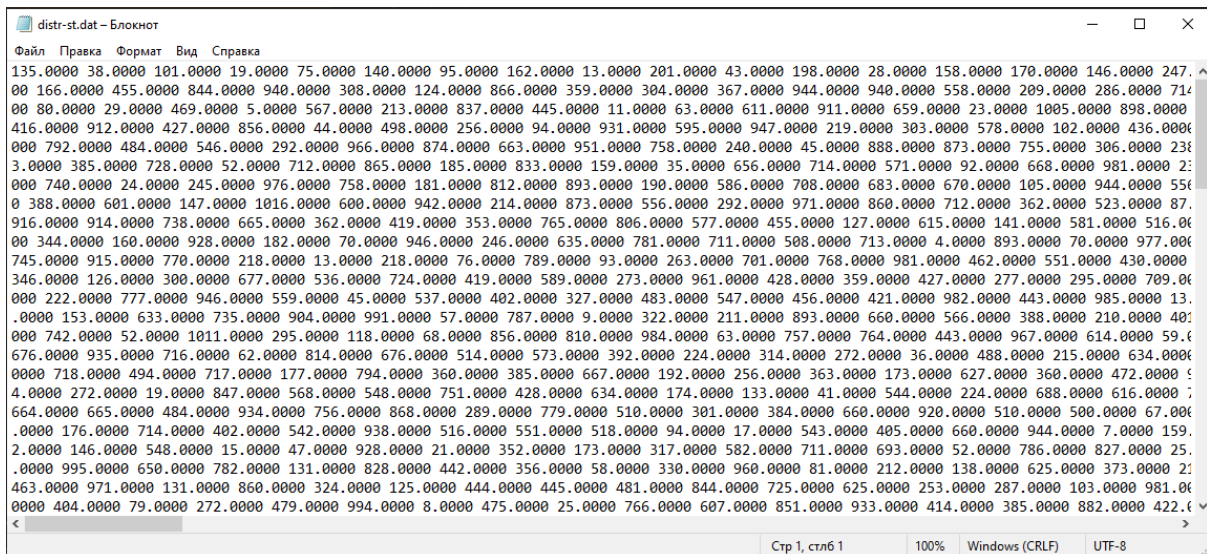
/d:st /p1:0 /p2:1 /f:out.txt

Исходный текст алгоритма:

```
def st(p1, p2, file_name):  
    x = []  
    with open(file_name) as f:  
        for line in f:  
            for a in line.split():  
                x.append(int(a))  
  
    res = []  
    left_border = 0  
    right_border = len(x)  
    for i in range(left_border, right_border):
```

```
    u = x[i] % 1024
    y = p2 * u + p1
    y = format(y, '.4f')
    res.append(y)
return res
```

Результат работы программы:



2. Треугольное распределение

Описание алгоритма:

$$\text{Функция распределения } a \leq x < a + b: f(x) = \frac{b - |a - x|}{b^2}$$

Если стандартные случайные числа $U1$ и $U2$ независимо получены методом генерации стандартного равномерного числа, то случайное число Y , подчиняющееся треугольному распределению, определяют по формуле:

$$Y = a + b * (U1 + U2 - 1).$$

Параметры запуска программы:

/d:tr /p1:1 /p2:2 /f:out.txt

Результат работы программы:

```
def tr(p1, p2, file_name):  
    x = []  
    with open(file_name) as f:  
        for line in f:  
            for a in line.split():  
                x.append(int(a))  
    res = []  
    left_border = 1  
    right_border = len(x)  
    for i in range(left_border, right_border):
```

```

u1 = x[i - 1] % 1024

u2 = x[i] % 1024

y = p1 + p2 * (u1 + u2 - 1)

y = format(y, '.4f')

res.append(y)


return res

```

Результат работы программы:

```

distr-tr.d - Блокнот
Файл  Правка  Формат  Вид  Справка
0.345.0000 0.277.0000 0.239.0000 0.187.0000 0.429.0000 0.469.0000 0.513.0000 0.349.0000 0.427.0000 0.487.0000 0.481.0000 0.451.0000 0.371.0000 0.655.0000 0.631.0000 0.785.0000 0.687.0000 0.773.0000 0.725.0000 0.595.0000
0.1319.0000 0.599.0000 0.171.0000 0.1039.0000 0.2269.0000 0.1755.0000 0.953.0000 0.1071.0000 0.2089.0000 0.1869.0000 0.1241.0000 0.2597.0000 0.3567.0000 0.2495.0000 0.863.0000 0.1979.0000 0.2449.0000 0.1325.0000 0.2613.0000
0.2807.0000 0.581.0000 0.1217.0000 0.1815.0000 0.1755.0000 0.1223.0000 0.509.0000 0.649.0000 0.2281.0000 0.2853.0000 0.1885.0000 0.783.0000 0.1017.0000 0.1533.0000 0.2449.0000 0.2241.0000 0.1275.0000 0.2865.0000
0.1585.0000 0.2279.0000 0.1899.0000 0.301.0000 0.1183.0000 0.1203.0000 0.903.0000 0.2479.0000 0.2543.0000 0.1971.0000 0.2711.0000 0.1763.0000 0.1803.0000 0.2309.0000 0.1787.0000 0.1209.0000 0.110.0000
0.2137.0000 0.1769.0000 0.1573.0000 0.1895.0000 0.2113.0000 0.1853.0000 0.1845.0000 0.2787.0000 0.2537.0000 0.2109.0000 0.2033.0000 0.1967.0000 0.1851.0000 0.925.0000 0.549.0000 0.901.0000 0.1917.0000 0.19.0000
0.1575.0000 0.371.0000 0.569.0000 0.1775.0000 0.3233.0000 0.2573.0000 0.2117.0000 0.3361.0000 0.3205.0000 0.1903.0000 0.1449.0000 0.2579.0000 0.2169.0000 0.1113.0000 0.1173.0000 0.2561.0000 0.3057.0000 0.1.0000
0.2387.0000 0.2793.0000 0.1741.0000 0.1765.0000 0.1087.0000 0.1797.0000 0.3081.0000 0.3587.0000 0.2515.0000 0.1721.0000 0.1437.0000 0.215.0000 0.1043.0000 0.2607.0000 0.3047.0000 0.3457.0000 0.3219.0000 0.20.0000
0.1153.0000 0.1817.0000 0.1707.0000 0.647.0000 0.2227.0000 0.3683.0000 0.1913.0000 0.341.0000 0.1399.0000 0.2401.0000 0.1909.0000 0.1187.0000 0.1185.0000 0.2621.0000 0.3055.0000 0.1743.0000 0.2327.00.0000
0.2659.0000 0.2079.0000 0.1403.0000 0.2641.0000 0.2639.0000 0.1591.0000 0.2869.0000 0.2441.0000 0.2017.0000 0.2355.0000 0.1739.0000 0.1947.0000 0.1489.0000 0.2371.0000 0.3075.0000 0.1249.0000 0.1929.0000
0.2801.0000 0.3787.0000 0.2925.0000 0.2485.0000 0.3427.0000 0.1901.0000 0.869.0000 0.2815.0000 0.3167.0000 0.2347.0000 0.1587.0000 0.919.0000 0.1981.0000 0.3393.0000 0.2903.0000 0.2829.0000 0.1873.0000 0.93.0000
0.2031.0000 0.2383.0000 0.1761.0000 0.2831.0000 0.2983.0000 0.2437.0000 0.2441.0000 0.1433.0000 0.1793.0000 0.1925.0000 0.2093.0000 0.3849.0000 0.2839.0000 0.1919.0000 0.1799.0000 0.2247.0000 0.1973.0000 0.039.0000
0.3121.0000 0.3533.0000 0.3091.0000 0.2005.0000 0.1909.0000 0.2217.0000 0.1977.0000 0.2373.0000 0.3319.0000 0.3369.0000 0.1975.0000 0.461.0000 0.461.0000 0.587.0000 0.1729.0000 0.1763.0000 0.711.0000 0.1927.00.0000
0.1983.0000 0.69.0000 0.1445.0000 0.2579.0000 0.2983.0000 0.2437.0000 0.2441.0000 0.1433.0000 0.1793.0000 0.1925.0000 0.2093.0000 0.3849.0000 0.2839.0000 0.1919.0000 0.1799.0000 0.2247.0000 0.1973.0000 0.039.0000
0.1621.0000 0.1707.0000 0.3141.0000 0.2871.0000 0.2623.0000 0.2183.0000 0.1343.0000 0.1217.0000 0.789.0000 0.1445.0000 0.1509.0000 0.2045.0000 0.3167.0000 0.2987.0000 0.3145.0000 0.2277.0000 0.759.0000 0.138.0000
0.1977.0000 0.757.0000 0.635.0000 0.2417.0000 0.3269.0000 0.2213.0000 0.1147.0000 0.1581.0000 0.2849.0000 0.2175.0000 0.1841.0000 0.1347.0000 0.1041.0000 0.1599.0000 0.885.0000 0.889.0000 0.1533.0000 0.27.0000
0.1507.0000 0.1131.0000 0.2509.0000 0.3101.0000 0.2341.0000 0.1399.0000 0.1043.0000 0.1141.0000 0.1353.0000 0.2253.0000 0.3217.0000 0.3217.0000 0.2077.0000 0.2099.0000 0.2963.0000 0.1849.0000 0.1993.0000 0.47.0000
0.2715.0000 0.2519.0000 0.2831.0000 0.2329.0000 0.1625.0000 0.1871.0000 0.1283.0000 0.2181.0000 0.1993.0000 0.1033.0000 0.1243.0000 0.1897.0000 0.1627.0000 0.1617.0000 0.2629.0000 0.2287.0000 0.2401.00.0000
0.3201.0000 0.2981.0000 0.2445.0000 0.1609.0000 0.705.0000 0.1657.0000 0.2929.0000 0.1997.0000 0.943.0000 0.2327.0000 0.2187.0000 0.1953.0000 0.3441.0000 0.3765.0000 0.1909.0000 0.1007.0000 0.1879.0000 0.9.0000
0.1201.0000 0.1303.0000 0.1083.0000 0.2229.0000 0.2217.0000 0.2509.0000 0.1309.0000 0.1791.0000 0.1575.0000 0.1427.0000 0.2823.0000 0.3441.0000 0.2355.0000 0.1209.0000 0.1317.0000 0.603.0000 0.363.0000 0.893.0000
0.2885.0000 0.1423.0000 0.2399.0000 0.3865.0000 0.2843.0000 0.1345.0000 0.2539.0000 0.2883.0000 0.1089.0000 0.331.0000 0.1055.0000 0.1945.0000 0.1901.0000 0.2079.0000 0.1879.0000 0.2025.0000 0.2051.0000 0.2.0000
0.2577.0000 0.1621.0000 0.1369.0000 0.2087.0000 0.3159.0000 0.2859.0000 0.2019.0000 0.1133.0000 0.1161.0000 0.2517.0000 0.2121.0000 0.2463.0000 0.2915.0000 0.2987.0000 0.3637.0000 0.2635.0000 0.1203.0.0000
0.2313.0000 0.2423.0000 0.3215.0000 0.3459.0000 0.2619.0000 0.975.0000 0.1779.0000 0.2231.0000 0.1887.0000 0.2959.0000 0.2907.0000 0.2133.0000 0.2137.0000 0.1223.0000 0.221.0000 0.1119.0000 0.1895.0000 0.3.0000
0.2723.0000 0.3677.0000 0.2000 0.2069.0000 0.561.0000 0.1383.0000 0.2059.0000 0.1249.0000 0.1839.0000 0.2923.0000 0.2359.0000 0.1905.0000 0.2833.0000 0.2065.0000 0.1467.0000 0.3125.0000 0.2547.0000 0.1407.0000 0.18.0000
0.2313.0000 0.2299.0000 0.3459.0000 0.1861.0000 0.1533.0000 0.2031.0000 0.1651.0000 0.2259.0000 0.1859.0000 0.2157.0000 0.2303.0000 0.2375.0000 0.3479.0000 0.2957.0000 0.1851.0000 0.2069.0000 0.1505.0000 0.7.0000
0.2543.0000 0.1821.0000 0.2807.0000 0.3977.0000 0.3077.0000 0.2801.0000 0.2501.0000 0.1557.0000 0.1121.0000 0.1789.0000 0.2341.0000 0.1527.0000 0.1961.0000 0.2255.0000 0.1435.0000 0.2371.0000 0.3015.0000 0.000.0000
0.1237.0000 0.1133.0000 0.2917.0000 0.3709.0000 0.1741.0000 0.2049.0000 0.3785.0000 0.3651.0000 0.3711.0000 0.3157.0000 0.1715.0000 0.1949.0000 0.2865.0000 0.3239.0000 0.2747.0000 0.2000 0.2577.0000 0.3689.0000 0.3.0000
0.2305.0000 0.3203.0000 0.3585.0000 0.1965.0000 0.2237.0000 0.2983.0000 0.2641.0000 0.2355.0000 0.2623.0000 0.2791.0000 0.1715.0000 0.1761.0000 0.1745.0000 0.2147.0000 0.2941.0000 0.2915.0000 0.2505.00.0000
0.2945.0000 0.3587.0000 0.3523.0000 0.3445.0000 0.2433.0000 0.2149.0000 0.2635.0000 0.2703.0000 0.1745.0000 0.1315.0000 0.1237.0000 0.213.0000 0.515.0000 0.1137.0000 0.2575.0000 0.3255.0000 0.1785.0000 0.1773.0000
0.2085.0000 0.2685.0000 0.2669.0000 0.897.0000 0.1151.0000 0.2203.0000 0.2221.0000 0.2049.0000 0.1247.0000 0.2089.0000 0.2327.0000 0.1939.0000 0.2299.0000 0.1549.0000 0.2531.0000 0.2109.0000 0.659.0000 0.1317.00.0000
0.881.0000 0.677.0000 0.1419.0000 0.2989.0000 0.3557.0000 0.3859.0000 0.2647.0000 0.2555.0000 0.2659.0000 0.2111.0000 0.2275.0000 0.1345.0000 0.603.0000 0.1535.0000 0.2659.0000 0.2047.0000 0.2433.0000 0.3451.0000
0.2227.0000 0.1165.0000 0.1493.0000 0.2911.0000 0.3565.0000 0.1877.0000 0.1633.0000 0.3145.0000 0.2309.0000 0.1221.0000 0.1651.0000 0.1317.0000 0.2277.0000 0.2591.0000 0.2607.0000 0.3613.0000 0.2613.0000
0.3083.0000 0.2417.0000 0.2143.0000 0.2637.0000 0.2731.0000 0.1075.0000 0.149.0000 0.525.0000 0.1529.0000 0.1783.0000 0.875.0000 0.903.0000 0.1759.0000 0.1525.0000 0.925.0000 0.993.0000 0.915.0000 0.689.0000
0.1481.0000 0.1409.0000 0.1733.0000 0.881.0000 0.447.0000 0.273.0000 0.989.0000 0.2491.0000 0.2269.0000 0.2111.0000 0.2445.0000 0.1051.0000 0.2087.0000 0.2037.0000 0.1829.0000 0.2941.0000 0.1657.0000 0.2509.0000 0.2201.0000
0.1201.0000 0.967.0000 0.1077.0000 0.857.0000 0.2501.0000 0.2349.0000 0.1319.0000 0.2725.0000 0.3409.0000 0.1979.0000 0.1271.0000 0.2787.0000 0.2359.0000 0.1407.0000 0.1791.0000 0.1035.0000 0.533.0000 0.95.0000
0.3361.0000 0.3811.0000 0.2609.0000 0.1705.0000 0.1425.0000 0.2465.0000 0.3849.0000 0.2811.0000 0.1845.0000 0.2073.0000 0.3109.0000 0.3319.0000 0.2985.0000 0.3529.0000 0.3423.0000 0.1565.0000 0.1039.00.0000
0.1975.0000 0.1211.0000 0.1383.0000 0.2315.0000 0.2939.0000 0.3033.0000 0.3377.0000 0.3415.0000 0.2255.0000 0.1305.0000 0.2537.0000 0.3775.0000 0.2627.0000 0.2595.0000 0.2529.0000 0.1897.0000 0.2551.00.0000
0.3695.0000 0.2829.0000 0.1553.0000 0.525.0000 0.841.0000 0.1825.0000 0.2455.0000 0.2039.0000 0.739.0000 0.1337.0000 0.1685.0000 0.625.0000 0.467.0000 0.1091.0000 0.2579.0000 0.3715.0000 0.3895.0000 0.2.0000
0.969.0000 0.1737.0000 0.1259.0000 0.1353.0000 0.1181.0000 0.539.0000 0.579.0000 0.411.0000 0.983.0000 0.763.0000 0.1805.0000 0.2289.0000 0.1615.0000 0.2895.0000 0.3377.0000 0.2907.0000 0.1339.0000 0.307.00.0000
0.1507.0000 0.1805.0000 0.2373.0000 0.1931.0000 0.3039.0000 0.2365.0000 0.1465.0000 0.2651.0000 0.1903.0000 0.1623.0000 0.1939.0000 0.663.0000 0.355.0000 0.1427.0000 0.1327.0000 0.2027.0000 0.2363.0000 0.603.0000
0.601.0000 0.987.0000 0.2555.0000 0.3253.0000 0.2457.0000 0.2261.0000 0.1879.0000 0.1427.0000 0.1441.0000 0.1349.0000 0.1037.0000 0.779.0000 0.1421.0000 0.2011.0000 0.1251.0000 0.959.0000 0.1621.0.0000
0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000 0.000.0000

```

3. Общее экспоненциальное распределение

Описание алгоритма:

Функция распределения $x \geq a, f(x) = \frac{1}{b} * \exp(-\frac{x-a}{b})$

Случайное число, соответствующее экспоненциальному распределению, получают по формуле: $Y = -b * \ln(U) + a$.

Параметры запуска программы:

/d:ex /p1:1 /p2:2 /f:out.txt

Исходный текст алгоритма:

```
def ex(p1, p2, file_name):  
    x = []  
    with open(file_name) as f:  
        for line in f:  
            for a in line.split():  
                x.append(int(a))  
  
    res = []  
    left_border = 0  
    right_border = len(x)  
    for i in range(left_border, right_border):  
        u = x[i] % 1024  
        y = math.log1p(u) * (-p2) + p1  
        y = format(y, '.4f')  
        res.append(y)  
  
    return res
```


Пример запуска программы.

dist-dir.nat - Блокнот

Файл

Правка

Формат

Вид

Справка

7.2644 1.0000 0.7663 1.0000 6.8771 1.0000 7.0358 1.0000 5.5299 1.0000 6.4854 1.0000 6.1631 1.0000 7.4099 1.0000 7.6389 1.0000 7.7349 1.0000 7.5825 1.0000 7.5958 1.0000 7.6165 1.0000 7.9333 1.0000 8.1518 1.0000 8.7818 1.0000 8.2876 1.0000 7.7430 1.0000 8.7540 1.0000 8.3465 1.0000 7.7725 1.0000 8.2394 1.0000 8.8791 1.0000 8.4273 1.0000 8.2449 1.0000 8.5955 1.0000 8.2402 1.0000 8.4158 1.0000 8.2085 1.0000 5.5299 1.0000 7.9729 1.0000 8.0372 1.0000 6.3332 1.0000 8.0170 1.0000 8.1792 1.0000 8.1471 1.0000 8.1049 1.0000 8.2290 1.0000 8.0097 1.0000 7.9514 1.0000 5.0787 1.0000 8.0158 1.0000 8.0278 1.0000 7.8218 1.0000 8.3243 1.0000 5.0886 1.0000 8.1259 1.0000 6.9924 1.0000 7.7813 1.0000 6.3332 1.0000 8.2503 1.0000 7.0145 1.0000 8.4235 1.0000 8.1620 1.0000 8.2534 1.0000 8.2685 1.0000 8.0073 1.0000 8.3805 1.0000 7.7370 1.0000 7.5404 1.0000 8.2562 1.0000 8.2093 1.0000 7.2548 1.0000 8.3529 1.0000 8.6771 1.0000 8.1308 1.0000 7.5548 1.0000 8.3964 1.0000 7.7076 1.0000 8.0422 1.0000 8.4146 1.0000 6.3747 1.0000 8.2394 1.0000 7.9543 7.6011 1.0000 8.2503 1.0000 8.2274 1.0000 7.9417 1.0000 8.3148 1.0000 7.3836 1.0000 7.5196 1.0000 7.8183 1.0000 7.9716 1.0000 7.9487 1.0000 8.2610 1.0000 8.3296 1.0000 6.3545 3516 1.0000 8.3573 1.0000 7.4523 1.0000 8.3491 1.0000 8.2861 1.0000 8.1069 1.0000 6.9999 1.0000 7.3719 1.0000 7.4523 1.0000 8.0489 1.0000 8.1853 1.0000 8.2480 1.0000 8.2495 1.0000 1.0 1.0000 8.1845 1.0000 8.3894 1.0000 8.2127 1.0000 8.2752 1.0000 7.7609 1.0000 8.3107 1.0000 8.1140 1.0000 8.3817 1.0000 6.4939 1.0000 8.3446 1.0000 8.3598 1.0000 8.2803 1.0000 1.0000 8.2926 1.0000 7.6967 1.0000 7.3054 1.0000 8.2346 1.0000 7.2099 1.0000 7.9923 1.0000 7.4135 1.0000 7.5958 1.0000 7.9663 1.0000 7.6190 1.0000 8.0361 1.0000 6.8200 1.0000 8.3038 1.0000 8.1720 1.0000 8.0122 1.0000 7.1322 1.0000 7.9676 1.0000 8.1546 1.0000 8.3900 1.0000 7.6165 1.0000 8.2693 1.0000 8.4322 1.0000 8.3309 1.0000 8.7349 1.0000 6.6223 1.0000 7.7451 1.0000 8.6180 1.0000 8.3196 1.0000 7.5772 1.0000 8.2876 1.0000 8.4158 1.0000 7.9807 1.0000 8.5943 1.0000 8.3327 1.0000 8.1299 1.0000 6.557 1.0000 8.2184 1.0000 8.2796 1.0000 7.7471 1.0000 8.3959 1.0000 8.2101 1.0000 7.5462 1.0000 8.3446 1.0000 7.7950 1.0000 7.6486 1.0000 8.4311 1.0000 7.7417 9676 1.0000 8.0861 1.0000 8.0956 1.0000 8.0011 1.0000 8.2752 1.0000 6.3747 1.0000 8.5955 1.0000 5.9728 1.0000 8.2233 1.0000 8.3703 1.0000 8.3800 1.0000 7.7838 1.0000 7.9116 1.0000 94 1.0000 8.0146 1.0000 7.9569 1.0000 7.3396 1.0000 8.0023 1.0000 6.7190 1.0000 7.9555 1.0000 7.8440 1.0000 8.0085 1.0000 6.7460 1.0000 8.3906 1.0000 8.2110 1.0000 8.3970 1.0000 1.0000 7.9304 1.0000 8.4028 1.0000 7.3680 1.0000 8.4079 1.0000 8.1949 1.0000 8.0048 1.0000 7.9514 1.0000 8.2737 1.0000 3.3548 1.0000 7.3142 1.0000 6.8950 1.0000 7.6389 1.0000 7.6315 1.0000 8.4158 1.0000 7.9333 1.0000 8.1160 1.0000 5.7609 1.0000 8.4028 1.0000 7.5660 1.0000 8.2678 1.0000 8.3003 1.0000 8.0266 1.0000 6.4682 1.0000 7.5984 1.0000 8.00 8.3855 1.0000 7.8951 1.0000 8.0442 1.0000 8.4268 1.0000 8.0914 1.0000 7.9042 1.0000 7.4062 1.0000 7.9846 1.0000 8.2085 1.0000 7.6510 1.0000 8.1289 1.0000 8.1593 1.0000 7.902 8.1966 1.0000 8.4224 1.0000 7.9923 1.0000 8.4440 1.0000 8.1220 1.0000 7.1381 1.0000 6.9123 1.0000 8.0266 1.0000 5.0787 1.0000 6.0745 1.0000 8.1602 1.0000 8.3964 1.0000 7.9012 8356 1.0000 7.8951 1.0000 7.5105 1.0000 8.1299 1.0000 8.3503 1.0000 7.5012 1.0000 8.4135 1.0000 8.1602 1.0000 7.5404 1.0000 8.2803 1.0000 8.2572 1.0000 8.3216 1.0000 8.2996 1.0000 63 1.0000 6.9999 1.0000 8.4360 1.0000 8.8200 1.0000 7.7225 1.0000 7.1723 1.0000 7.7308 1.0000 8.2076 1.0000 7.6674 1.0000 7.3438 1.0000 8.1259 1.0000 7.1555 1.0000 7.7550 1.0000 1.0000 8.4208 1.0000 8.1940 1.0000 8.0219 1.0000 8.6861 1.0000 7.9935 1.0000 7.5196 1.0000 7.8130 1.0000 8.3745 1.0000 8.2330 1.0000 7.6812 1.0000 7.9262 1.0000 8.0765 1.0000 0.000 8.3316 1.0000 3.3548 1.0000 7.4724 1.0000 7.7838 1.0000 7.8094 1.0000 7.5462 1.0000 7.6365 1.0000 7.2548 1.0000 8.1675 1.0000 8.2961 1.0000 8.4360 1.0000 7.9116 1.0000 8.00 8.

Случайное число Y , подчиняющееся гамма распределению, определяют по формуле: $Y = a - b \ln\{(1 - U_i)(1 - U_{i+1})\}$.

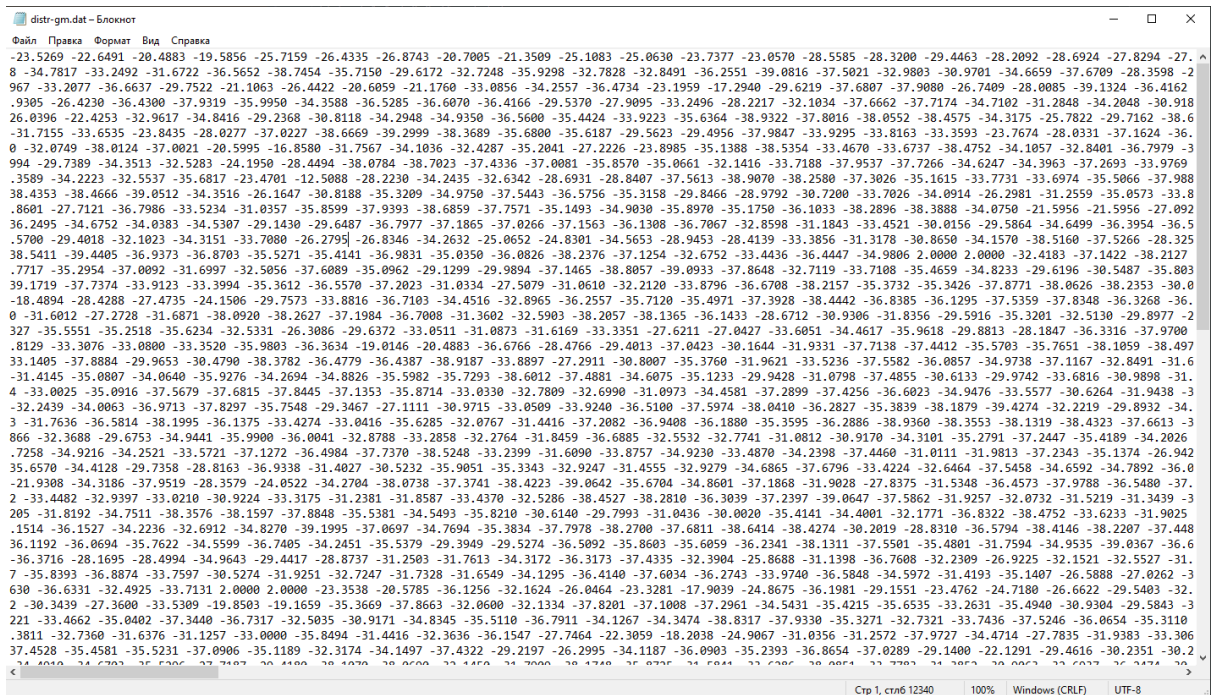
Параметры запуска программы:

/d:gm /p1:2 /p2:3 /f:out.txt

Исходный текст алгоритма:

```
def gm(p1, p2, p3, file_name):
    x = []
    with open(file_name) as f:
        for line in f:
            for a in line.split():
                x.append(int(a))
    res = []
    left_border = 1
    right_border = len(x)
    for i in range(left_border, right_border):
        u = []
        for j in range(p3):
            u.append(x[i - j] % 1024)
        u_mult = 1
        for j in range(len(u)):
            u_mult = u_mult * (1 - u[j])
        y = p1 - p2 * math.log1p(math.fabs(u_mult))
        y = format(y, '.4f')
        res.append(y)
    return res
```

Результат работы программы:



6. Логнормальное распределение

Описание алгоритма:

Функция распределения $f(x) = \frac{1}{\left(\sqrt{2\pi}\left(\frac{x-a}{b}\right)\right)\text{exprexp}\left\{-\frac{1}{2}\left(\frac{x-a}{b}\right)^2\right\}} \geq a$

Преобразование:

$y1, y2 = \text{lognorm}(x1, x2, a, b): z1, z2 = \text{norm}(x1, x2, 0, 1, m) y1 = a + \exp(b - z1)$

Случайное число Y , подчиняющееся логнормальному распределению, определяют по формуле: $Y = a + \exp(b - U)$.

Параметры запуска программы:

/d:ln /p1:1 /p2:2 /f:out.txt

Исходный текст алгоритма:

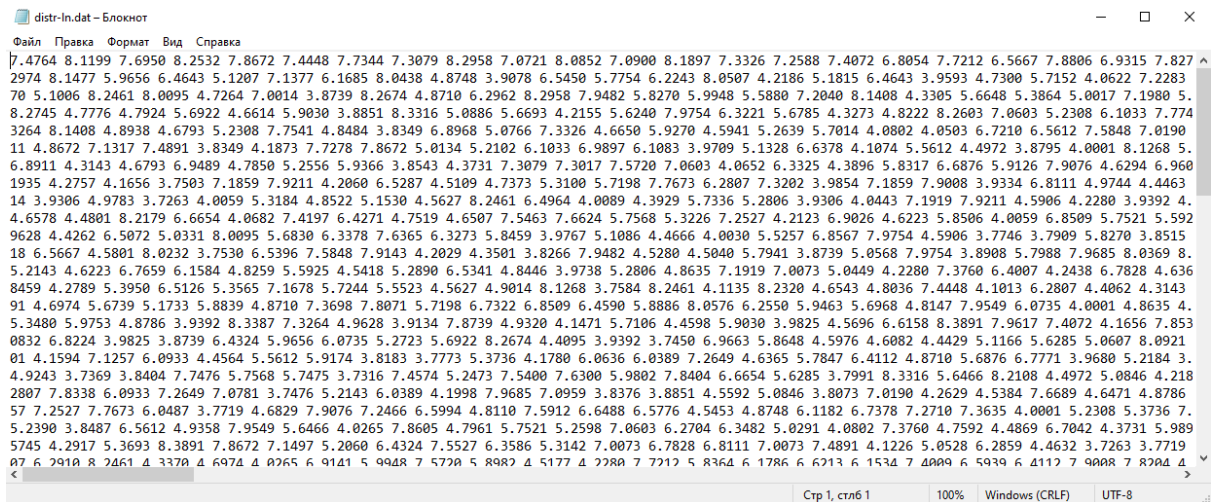
```
def ln(p1, p2, file_name):
    x = []
    with open(file_name) as f:
        for line in f:
            for a in line.split():
                x.append(int(a))
    res = []
```

```

left_border = 0
right_border = len(x)
for i in range(left_border, right_border):
    u = x[i] % 1024
    y = math.exp(p2 - u) + p1
    y = format(y, '.4f')
    res.append(y)
return res

```

Результат работы программы:



дistr-ln.dat - Блокнот

Файл Правка Формат Вид Справка

7.4764 8.1199 7.6950 8.2532 7.8672 7.4448 7.7344 7.3079 8.2958 7.0721 8.0852 7.0900 8.1897 7.3326 7.2588 7.4072 6.8054 7.7212 6.5667 7.8806 6.9315 7.827
2974 8.1477 5.9656 6.4643 5.1207 7.1377 6.1685 8.0438 4.8748 3.9078 6.5450 5.7754 6.2243 8.0507 4.2186 5.1815 6.4643 3.9593 4.7300 5.7152 4.0622 7.2283
70 5.1006 8.2461 8.0095 4.7264 7.0014 3.8739 8.2674 4.8710 6.2962 8.2958 7.9482 5.8270 5.9948 5.5880 7.2040 8.1408 4.3305 5.6648 5.3864 5.0017 7.1980 5.
8.2745 4.7776 4.7924 5.6922 4.6614 5.9030 3.8851 8.3316 5.0886 5.6693 4.2155 5.6240 7.9754 6.3221 5.6785 4.3273 4.8222 8.2603 7.0603 5.2308 6.1033 7.774
3264 8.1408 4.8938 4.6793 5.2308 7.7541 4.8484 3.8349 6.8968 5.0766 7.3326 4.6650 5.9270 4.5941 5.2639 5.7014 4.0802 4.0503 6.7210 6.5612 7.5848 7.0190
11 4.8672 7.1317 7.4891 3.8349 4.1873 7.7278 7.8672 5.0134 5.2102 6.1033 6.9897 6.1083 3.9709 5.1328 6.6378 4.1074 5.5612 4.4972 3.8795 4.0001 8.1268 5.
6.8911 4.3143 4.6793 6.9489 4.7850 5.2556 5.9366 3.8543 4.3731 7.3079 7.3017 7.5720 7.0603 4.0652 6.3325 4.3896 5.8317 6.6876 5.9126 7.9076 4.6294 6.960
1935 4.2757 4.1656 3.7503 7.1859 7.9211 4.2060 6.5287 4.5109 4.7373 5.3100 5.7198 7.7673 6.2807 7.3202 3.9854 7.1859 7.9008 3.9334 6.8111 4.9744 4.4463
14 3.9306 4.9783 3.7263 4.0059 5.3184 4.8522 5.1530 4.5627 8.2461 6.4964 4.0089 4.3929 5.7336 5.2806 3.9306 4.0443 7.1919 7.9211 4.5906 4.2280 3.9392 4.
4.6578 4.4801 8.2179 6.6654 4.0682 7.4197 6.4271 4.7519 4.6507 7.5463 7.6624 5.7568 5.3226 7.2527 4.2123 6.9026 4.6223 5.8506 4.0059 6.8509 5.7521 5.592
9628 4.4262 6.5072 5.0331 8.0095 5.6830 6.3378 7.6365 6.3273 5.8459 3.9767 5.1086 4.4666 4.0030 5.5257 6.8567 7.9754 4.5906 3.7746 3.7909 5.8270 3.8515
18 6.5667 4.5801 8.0232 3.7530 6.5396 7.5848 7.9143 4.2029 4.3501 3.8266 7.9482 4.5280 4.5040 5.7941 3.8739 5.0568 7.9754 3.8908 5.7988 7.9685 8.0369 8.
5.2143 4.6223 6.7659 6.1584 4.8259 5.5925 4.5418 5.2890 6.5341 4.8446 3.9738 5.2806 4.8635 7.1919 7.0073 5.0449 4.2280 7.3760 6.4007 4.2438 6.7828 4.636
8459 4.2789 5.3950 6.5126 5.3565 7.1678 5.7244 5.5523 4.5627 4.9014 8.1268 3.7584 8.2461 4.1135 8.2320 4.6543 4.8036 7.4448 4.1013 6.2807 4.4062 4.3143
91 4.6974 5.6739 5.1733 5.8839 4.8710 7.3698 7.8071 5.7198 6.7322 6.8509 6.4590 5.8886 8.0576 6.2550 5.9463 5.6968 4.8147 7.9549 6.0735 4.0001 4.8635 4.
5.3480 5.9753 4.8786 3.9392 8.3387 7.3264 4.9628 3.9134 7.8739 4.9320 4.1471 5.7106 4.4598 5.9030 3.9825 4.5696 6.6158 8.3891 7.9617 7.4072 4.1656 7.853
0832 6.8224 3.9825 3.8739 6.4324 5.9656 6.0735 5.2723 5.6922 8.2674 4.4095 3.9392 3.7450 6.9663 5.8648 4.5976 4.6082 4.4429 5.1166 5.6285 5.0607 8.0921
01 4.1594 7.1257 6.0933 4.4564 5.5612 5.9174 3.8183 3.7773 5.3736 4.1780 6.0636 6.0389 7.2649 4.6365 5.7847 6.4112 4.8710 5.6876 6.7771 3.9680 5.2184 3.
4.9243 3.7369 3.8404 7.7476 5.7568 5.7475 3.7316 7.4574 5.2473 7.5400 7.6300 5.9802 7.8404 6.6654 5.6285 3.7991 8.3316 5.6466 8.2108 4.4972 5.0846 4.218
2807 7.8338 6.0933 7.2649 7.0781 3.7476 5.2143 6.0389 4.1998 7.9685 7.0959 3.8376 3.8851 4.5592 5.0846 3.8073 7.0190 4.2629 4.5384 7.6689 4.6471 4.8786
57 7.2527 7.7673 6.0487 3.7719 4.6829 7.9076 7.2466 6.5994 4.8110 7.5912 6.6488 6.5776 4.5453 4.8748 6.1182 6.7378 7.2710 7.3635 4.0001 5.2308 5.3736 7.
5.2390 3.8487 6.5612 4.9358 7.9549 5.6466 4.0265 7.8605 4.7961 5.7521 5.2598 7.0603 6.2704 6.3482 5.0291 4.0802 7.3760 4.7592 4.4869 6.7042 4.3731 5.989
5745 4.2917 5.3693 8.3891 7.8672 7.1497 5.2060 6.4324 7.5527 6.3586 5.3142 7.0073 6.7828 6.8111 7.0073 7.4891 4.1226 5.0528 6.2859 4.4632 3.7263 3.7719
07 6 2910 8 2461 4 3370 4 6974 4 0765 6 9141 5 9948 7 5720 5 8982 4 5177 4 2280 7 7212 5 8364 6 1786 6 6213 6 1534 7 4009 6 5939 6 4112 7 9098 7 8704 4

Стр 1, столб 1 100% Windows (CRLF) UTF-8

7. Логистическое распределение

Описание алгоритма:

Случайное число Y , подчиняющееся логистическому распределению, определяют по формуле:

$$Y = a + b * \ln\left(\frac{U}{1-U}\right).$$

Параметры запуска программы:

/d:ls /p1:1 /p2:2 /f:out.txt

Исходный текст алгоритма:

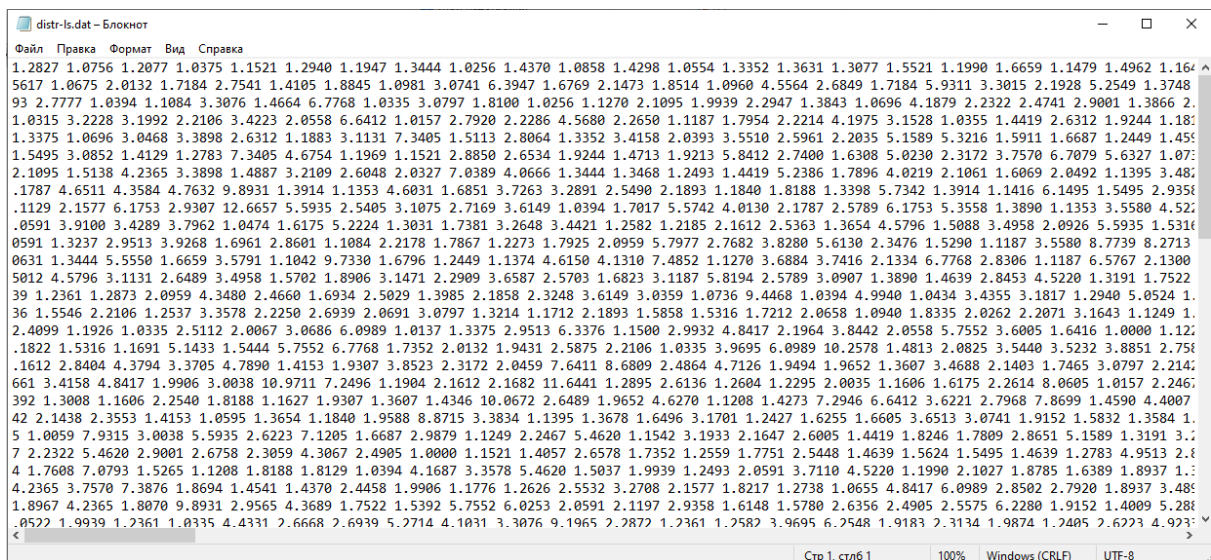
```
import math
```

```

def ls(p1, p2, file_name):
    x = []
    with open(file_name) as f:
        for line in f:
            for a in line.split():
                x.append(int(a))
    res = []
    left_border = 0
    right_border = len(x)
    for i in range(left_border, right_border):
        u = x[i] % 1024
        y = p1 + p2 * math.log1p(math.fabs(u / (1 - u)))
        y = format(y, '.4f')
        res.append(y)
    return res

```

Результат работы программы:



1.2827	1.0756	1.2077	1.0375	1.1521	1.2940	1.1947	1.3444	1.0256	1.4370	1.0858	1.4298	1.0554	1.3352	1.3631	1.3077	1.5521	1.1990	1.6659	1.1479	1.4962	1.164
5617	1.0675	2.0132	1.7184	2.7541	1.4105	1.8845	1.0981	3.0741	6.3947	1.6769	2.1473	1.8514	1.0960	4.5564	2.6849	1.7184	5.9311	3.3015	2.1928	5.2549	1.3748
93	2.7777	1.0394	1.1084	3.3076	1.4664	6.7768	1.0335	3.0797	1.8100	1.0256	1.1270	2.1095	1.9939	2.2947	1.3843	1.0696	4.1879	2.2322	2.4741	2.9001	1.3866
1.0315	3.2228	3.1992	2.2106	3.4223	2.0558	6.6412	1.0157	2.7920	2.2286	4.5680	2.2650	1.1187	1.7954	2.2214	4.1975	3.1528	1.0355	1.4419	2.6312	1.9244	1.181
1.3375	1.0696	3.0468	3.3898	2.6312	1.1883	3.1131	7.3405	1.5113	2.8064	1.3352	3.4158	2.0393	3.5510	2.5961	2.2035	5.1589	5.3216	1.5911	1.6687	1.2449	1.458
1.5495	3.0852	1.4129	1.2783	7.3405	4.6754	1.1969	1.1521	2.8850	2.6534	1.9244	1.4713	1.9213	5.8412	2.7400	1.6308	5.0230	2.3172	3.7570	6.7079	5.6327	1.075
2.1095	1.5138	4.2365	3.3898	1.4887	3.2109	2.6048	2.0327	7.0389	4.0666	1.3444	1.3468	1.2493	1.4419	5.2386	1.7896	4.0219	2.1061	1.6069	2.0492	1.1395	3.487
1.1787	4.6511	4.3584	4.7632	9.8931	1.3914	1.1353	4.6031	1.6851	3.7263	3.2891	2.5490	2.1893	1.1840	1.8188	1.3398	5.7342	1.3914	1.1416	6.1495	1.5495	2.9358
1.1129	2.1577	6.1753	2.9307	12.6657	5.5935	2.5405	3.1075	2.7169	3.6149	1.0394	1.7017	5.5742	4.0130	2.1787	2.5789	6.1753	5.3558	1.3890	1.1353	3.5580	4.527
1.0591	3.9100	3.4289	3.7962	1.0474	1.6175	5.2224	1.3031	1.7381	3.2648	3.4421	1.2582	1.2185	2.1612	2.5363	1.3654	4.5796	1.5088	3.4958	2.0926	5.5935	1.5316
0.591	1.3237	2.9513	3.9268	1.6961	2.8601	1.1084	2.2178	1.7867	1.2273	1.7925	2.0959	5.7977	2.7682	3.8280	5.6130	2.3476	1.5290	1.1187	3.5580	8.7739	8.2713
0.631	1.3444	5.5550	1.6659	3.5791	1.1042	9.7330	1.6796	1.2449	1.1374	4.6150	4.1310	7.4852	1.1270	3.6884	3.7416	2.1334	6.7768	2.8306	1.1187	6.5767	2.1300
5012	4.5796	3.1131	2.6489	3.4958	1.5702	1.8906	3.1471	2.2909	3.6587	2.5703	1.6823	3.1187	5.8194	2.5789	3.0907	1.3890	1.4639	2.8453	4.5220	1.3191	1.7522
39	1.2361	1.2873	2.0959	4.3480	2.4660	1.6934	2.5029	1.3985	2.1858	2.3248	3.6149	3.0359	1.0736	9.4468	1.0394	4.9940	1.0434	3.4355	3.1817	1.2940	5.0524
36	1.5546	2.2106	1.2537	3.3578	2.2250	2.6939	2.0691	3.0797	1.3214	1.1712	2.1893	1.5858	1.5316	1.7212	2.0658	1.0940	1.8335	2.0262	2.2071	3.1643	1.1249
2.4099	1.1926	1.0335	2.5112	2.0067	3.0686	6.0989	1.0137	1.3375	2.9513	6.3376	1.1500	2.9932	4.8417	2.1964	3.8442	2.0558	5.7552	3.6005	1.6416	1.0000	1.122
1.822	1.5316	1.1691	5.1433	1.5444	5.7552	6.7768	1.7352	2.0132	1.9431	2.5875	2.2106	1.0335	3.9695	6.0989	10.2578	1.4813	2.0825	3.5440	3.5232	3.8851	2.758
1.612	2.8404	4.3794	3.3705	4.7890	1.4153	1.9307	3.8523	2.3172	2.0459	7.6411	8.6809	2.4864	4.7126	1.9494	1.9652	1.3607	3.4688	2.1403	1.7465	3.0797	2.214
661	3.4158	4.8417	1.9906	3.0038	10.9711	7.2496	1.1904	2.1612	2.1682	11.6441	1.2895	2.6136	1.2604	1.2295	2.0035	1.1606	1.6175	2.2614	8.0605	1.0157	2.2467
392	1.3008	1.1606	2.2540	1.8188	1.1627	1.9307	1.3607	1.4346	10.0672	2.6489	1.9652	4.6270	1.1208	1.4273	7.2946	6.6412	3.6221	2.7968	7.8699	1.4590	4.4007
42	2.1438	2.3553	1.4153	1.0595	1.3654	1.1840	1.9588	8.8715	3.3834	1.1395	1.3678	1.6496	3.1701	1.2427	1.6255	1.6605	3.6513	3.0741	1.9152	1.5832	1.3584
5	1.0059	7.9315	3.0038	5.5935	2.6223	7.1205	1.6687	2.9879	1.1249	2.2467	5.4620	1.1542	3.1933	2.1647	2.6005	1.4419	1.8246	1.7809	2.8651	5.1589	1.3191
7	2.2322	5.4620	2.9001	2.6758	2.3059	4.3067	2.4905	1.0000	1.1521	1.4057	2.6578	1.7352	1.2559	1.7751	2.5448	1.4639	1.5624	1.5495	1.4639	1.2783	4.9513
4	1.7608	7.0793	1.5265	1.1208	1.8188	1.8129	1.0394	4.1687	3.3578	5.4620	1.5037	1.9939	1.2493	2.0591	3.7110	4.5220	1.1990	2.1027	1.8785	1.6389	1.8937
4.2365	3.7570	7.3876	1.8694	1.4541	1.4370	2.4458	1.9906	1.1776	1.2626	2.5532	3.2708	2.1577	1.8217	1.2738	1.0655	4.8417	6.0989	2.8502	2.7920	1.8937	3.488
1.8967	4.2365	1.8070	9.8931	2.9565	4.3689	1.7522	1.5392	5.7552	6.0253	2.0591	2.1197	2.9358	1.6148	1.5780	2.6356	2.4905	2.5575	6.2280	1.9152	1.4009	5.288
0.522	1.9939	1.2361	1.0335	4.4331	2.6668	7.6939	5.2714	4.1031	3.3076	9.1965	2.2872	1.2361	1.2582	3.9695	6.2548	1.9183	2.3134	1.9874	1.2405	2.6223	4.923

8. Биномиальное распределение

Описание алгоритма:

Преобразование:

$y = binominal(x, a, b, m): \quad u = U(x) \quad s = 0 \quad k = 0 \text{ loopstart:}$

$$s = s + Cbk ak (1 - a)b - k \quad \text{if } s > u: \quad y = k$$

Завершить

if $k < b - 1$: $k = k + 1$ перейти к loopstart

$y = b$

Коэффициент $C_b^k = \frac{b!}{k!(b-k)!}$

Параметры запуска программы:

/d:bi /p1:0.2 /p2:10 /f:out.txt

Исходный текст алгоритма:

```
def bi(p1, p2, file_name):
    x = []
    with open(file_name) as f:
        for line in f:
            for a in line.split():
                x.append(int(a))

    res = []
    left_border = 0
    right_border = len(x)
    for i in range(left_border, right_border):
        u = x[i] / 1024
        y = 0
        s = 0
        k = 0
        while (True):
            s = s + (math.factorial(p2) /
(math.factorial(k) * math.factorial(p2 - k))) * (p1 ** k) *\
                ((1 - p1) ** (p2 - k))
            if s > u:
                y = k
                break
            if k < p2 - 1:
                k = k + 1
                continue
```

```

        y = p2
        break
    res.append(y)
return res

```

Результат работы программы:

[illegible]

