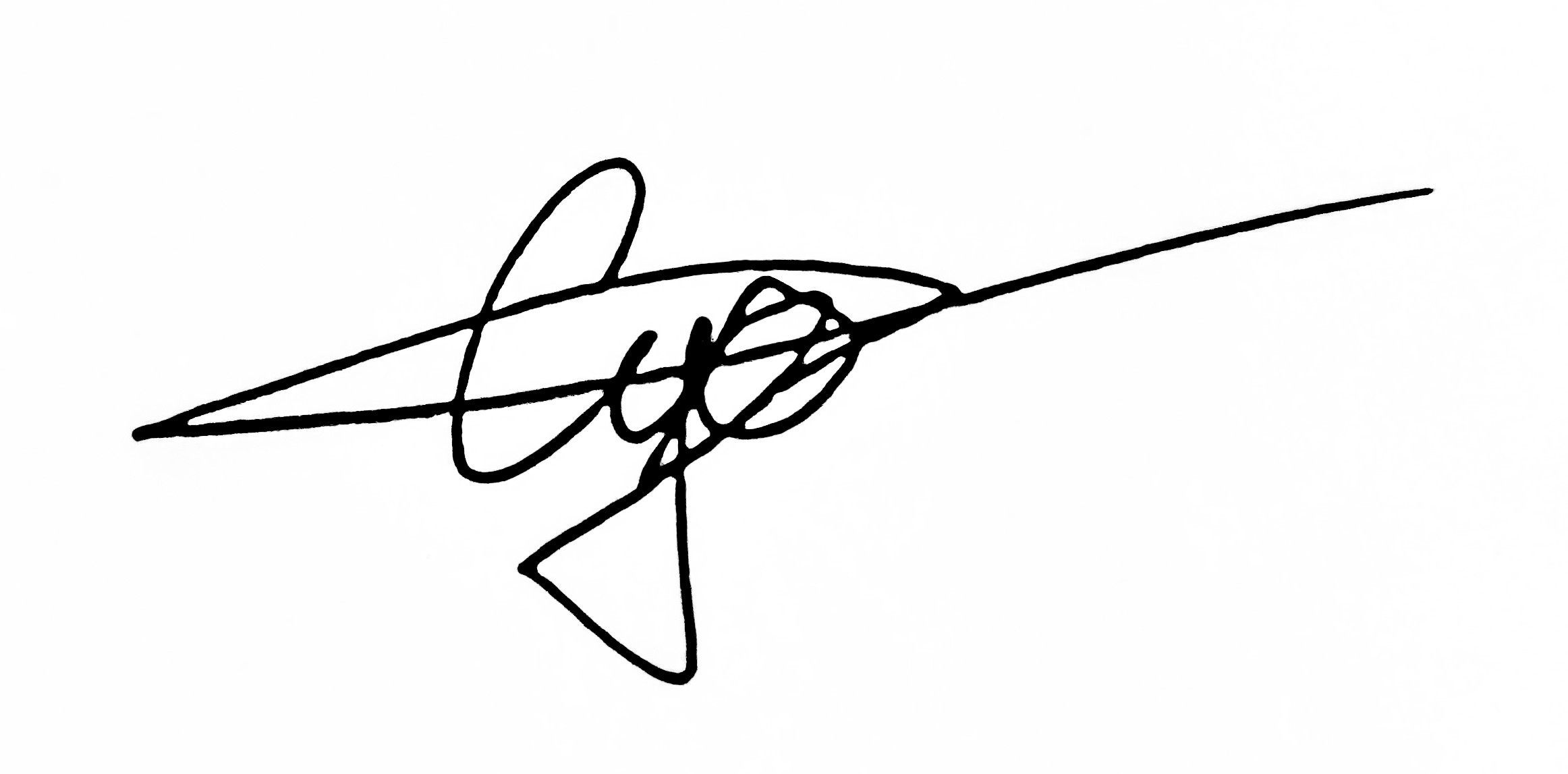
|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **ИУК «Информатика и управление»** |
| **КАФЕДРА** | **ИУК2 «Информационные системы и сети»** |
|  | |

**ЛАБОРАТОРНАЯ РАБОТА №9**

**«Моделирование физических процессов»**

**ДИСЦИПЛИНА: «Теоретическая информатика»**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил: студент гр. ИУК4-11Б | |  |  | ( | Суриков Н.С | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |
| Проверил: | |  |  | ( | Гладских А.П | ) |
|  |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |
| --- | --- |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | |
|  | - Балльная оценка:  - Оценка: |

**Цель:** получить последовательность из 1000 псевдослучайных чисел используя математические модели генерации случайных чисел.

**Задачи:**

* 1. Изучить теорию генерации псевдослучайных чисел
  2. Выполнить реализацию двух алгоритмов

**Ход решения:**

Для реализации были выбраны:

* Метод середины квадрата
* Аддитивный генератор Фибоначчи

*Листинг программы:*

1 def fibonacci\_generator(m, x, n):

2 for i in range(len(x) - 1, n):

3 next\_value = (x[i] + x[i - 1]) % m

4 x.append(next\_value)

5 return x

6

7

8 def middle\_square\_generator(seed, n):

9 random\_numbers = []

10 k = int(len(str(seed)))

11 for \_ in range(n):

12 square = seed \* seed

13 seed\_str = str(square).zfill(2 \* k)

14 middle = seed\_str[k // 2: -k // 2]

15 seed = int(middle)

16 random\_numbers.append(seed / 10\*\*k)

17 return random\_numbers

18

19

20 def main():

21 while True:

22 print("Выберите метод генерации псевдослучайных чисел:")

23 print("1. Аддитивный генератор Фибоначчи")

24 print("2. Метод середины квадрата")

25 print("0. Выход из программы")

26

27 choice = int(input("Введите ваш выбор: "))

28

29 match choice:

30 case 0:

31 print("Выход из программы")

32 return

33 case 1:

34 m = int(input("Введите m (четное число): "))

35 initial\_values = [

36 int(x)

37 for x in input(

38 "Введите начальные значения, разделенные пробелами: "

39 ).split()

40 ]

41 n = int(input("Введите количество генерируемых случайных чисел: "))

42 print(fibonacci\_generator(m, initial\_values, n))

43 case 2:

44 seed = int(input("Введите начальное значение: "))

45 n = int(input("Введите количество генерируемых случайных чисел: "))

46 print(middle\_square\_generator(seed, n))

47 case \_:

48 print("Неверный выбор")

49

50

51 if \_\_name\_\_ == "\_\_main\_\_":

52 main()

*Описание работы:*

Функция **fibonacci\_generator** реализует аддитивный генератор псевдослучайных чисел Фибоначчи. Она принимает параметры m, initial\_values и n, где m - модуль (должен быть четным), initial\_values - список начальных значений для последовательности, n - количество генерируемых псевдослучайных чисел. Функция возвращает сгенерированные псевдослучайные числа в виде списка.

Функция **middle\_square\_generator** реализует метод середины квадрата для генерации псевдослучайных чисел. Она принимает параметры seed и n, где seed - начальное значение для генерации, n - количество генерируемых псевдослучайных чисел. Функция возвращает сгенерированные псевдослучайные числа в виде списка.

*Результат работы:*

*Для аддитивного генератора Фибоначчи при m = 11942, начальных [1, 3]:*

[1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778, 9349, 3185, 592, 3777, 4369, 8146, 573, 8719, 9292, 6069, 3419, 9488, 965, 10453, 11418, 9929, 9405, 7392, 4855, 305, 5160, 5465, 10625, 4148, 2831, 6979, 9810, 4847, 2715, 7562, 10277, 5897, 4232, 10129, 2419, 606, 3025, 3631, 6656, 10287, 5001, 3346, 8347, 11693, 8098, 7849, 4005, 11854, 3917, 3829, 7746, 11575, 7379, 7012, 2449, 9461, 11910, 9429, 9397, 6884, 4339, 11223, 3620, 2901, 6521, 9422, 4001, 1481, 5482, 6963, 503, 7466, 7969, 3493, 11462, 3013, 2533, 5546, 8079, 1683, 9762, 11445, 9265, 8768, 6091, 2917, 9008, 11925, 8991, 8974, 6023, 3055, 9078, 191, 9269, 9460, 6787, 4305, 11092, 3455, 2605, 6060, 8665, 2783, 11448, 2289, 1795, 4084, 5879, 9963, 3900, 1921, 5821, 7742, 1621, 9363, 10984, 8405, 7447, 3910, 11357, 3325, 2740, 6065, 8805, 2928, 11733, 2719, 2510, 5229, 7739, 1026, 8765, 9791, 6614, 4463, 11077, 3598, 2733, 6331, 9064, 3453, 575, 4028, 4603, 8631, 1292, 9923, 11215, 9196, 8469, 5723, 2250, 7973, 10223, 6254, 4535, 10789, 3382, 2229, 5611, 7840, 1509, 9349, 10858, 8265, 7181, 3504, 10685, 2247, 990, 3237, 4227, 7464, 11691, 7213, 6962, 2233, 9195, 11428, 8681, 8167, 4906, 1131, 6037, 7168, 1263, 8431, 9694, 6183, 3935, 10118, 2111, 287, 2398, 2685, 5083, 7768, 909, 8677, 9586, 6321, 3965, 10286, 2309, 653, 2962, 3615, 6577, 10192, 4827, 3077, 7904, 10981, 6943, 5982, 983, 6965, 7948, 2971, 10919, 1948, 925, 2873, 3798, 6671, 10469, 5198, 3725, 8923, 706, 9629, 10335, 8022, 6415, 2495, 8910, 11405, 8373, 7836, 4267, 161, 4428, 4589, 9017, 1664, 10681, 403, 11084, 11487, 10629, 10174, 8861, 7093, 4012, 11105, 3175, 2338, 5513, 7851, 1422, 9273, 10695, 8026, 6779, 2863, 9642, 563, 10205, 10768, 9031, 7857, 4946, 861, 5807, 6668, 533, 7201, 7734, 2993, 10727, 1778, 563, 2341, 2904, 5245, 8149, 1452, 9601, 11053, 8712, 7823, 4593, 474, 5067, 5541, 10608, 4207, 2873, 7080, 9953, 5091, 3102, 8193, 11295, 7546, 6899, 2503, 9402, 11905, 9365, 9328, 6751, 4137, 10888, 3083, 2029, 5112, 7141, 311, 7452, 7763, 3273, 11036, 2367, 1461, 3828, 5289, 9117, 2464, 11581, 2103, 1742, 3845, 5587, 9432, 3077, 567, 3644, 4211, 7855, 124, 7979, 8103, 4140, 301, 4441, 4742, 9183, 1983, 11166, 1207, 431, 1638, 2069, 3707, 5776, 9483, 3317, 858, 4175, 5033, 9208, 2299, 11507, 1864, 1429, 3293, 4722, 8015, 795, 8810, 9605, 6473, 4136, 10609, 2803, 1470, 4273, 5743, 10016, 3817, 1891, 5708, 7599, 1365, 8964, 10329, 7351, 5738, 1147, 6885, 8032, 2975, 11007, 2040, 1105, 3145, 4250, 7395, 11645, 7098, 6801, 1957, 8758, 10715, 7531, 6304, 1893, 8197, 10090, 6345, 4493, 10838, 3389, 2285, 5674, 7959, 1691, 9650, 11341, 9049, 8448, 5555, 2061, 7616, 9677, 5351, 3086, 8437, 11523, 8018, 7599, 3675, 11274, 3007, 2339, 5346, 7685, 1089, 8774, 9863, 6695, 4616, 11311, 3985, 3354, 7339, 10693, 6090, 4841, 10931, 3830, 2819, 6649, 9468, 4175, 1701, 5876, 7577, 1511, 9088, 10599, 7745, 6402, 2205, 8607, 10812, 7477, 6347, 1882, 8229, 10111, 6398, 4567, 10965, 3590, 2613, 6203, 8816, 3077, 11893, 3028, 2979, 6007, 8986, 3051, 95, 3146, 3241, 6387, 9628, 4073, 1759, 5832, 7591, 1481, 9072, 10553, 7683, 6294, 2035, 8329, 10364, 6751, 5173, 11924, 5155, 5137, 10292, 3487, 1837, 5324, 7161, 543, 7704, 8247, 4009, 314, 4323, 4637, 8960, 1655, 10615, 328, 10943, 11271, 10272, 9601, 7931, 5590, 1579, 7169, 8748, 3975, 781, 4756, 5537, 10293, 3888, 2239, 6127, 8366, 2551, 10917, 1526, 501, 2027, 2528, 4555, 7083, 11638, 6779, 6475, 1312, 7787, 9099, 4944, 2101, 7045, 9146, 4249, 1453, 5702, 7155, 915, 8070, 8985, 5113, 2156, 7269, 9425, 4752, 2235, 6987, 9222, 4267, 1547, 5814, 7361, 1233, 8594, 9827, 6479, 4364, 10843, 3265, 2166, 5431, 7597, 1086, 8683, 9769, 6510, 4337, 10847, 3242, 2147, 5389, 7536, 983, 8519, 9502, 6079, 3639, 9718, 1415, 11133, 606, 11739, 403, 200, 603, 803, 1406, 2209, 3615, 5824, 9439, 3321, 818, 4139, 4957, 9096, 2111, 11207, 1376, 641, 2017, 2658, 4675, 7333, 66, 7399, 7465, 2922, 10387, 1367, 11754, 1179, 991, 2170, 3161, 5331, 8492, 1881, 10373, 312, 10685, 10997, 9740, 8795, 6593, 3446, 10039, 1543, 11582, 1183, 823, 2006, 2829, 4835, 7664, 557, 8221, 8778, 5057, 1893, 6950, 8843, 3851, 752, 4603, 5355, 9958, 3371, 1387, 4758, 6145, 10903, 5106, 4067, 9173, 1298, 10471, 11769, 10298, 10125, 8481, 6664, 3203, 9867, 1128, 10995, 181, 11176, 11357, 10591, 10006, 8655, 6719, 3432, 10151, 1641, 11792, 1491, 1341, 2832, 4173, 7005, 11178, 6241, 5477, 11718, 5253, 5029, 10282, 3369, 1709, 5078, 6787, 11865, 6710, 6633, 1401, 8034, 9435, 5527, 3020, 8547, 11567, 8172, 7797, 4027, 11824, 3909, 3791, 7700, 11491, 7249, 6798, 2105, 8903, 11008, 7969, 7035, 3062, 10097, 1217, 11314, 589, 11903, 550, 511, 1061, 1572, 2633, 4205, 6838, 11043, 5939, 5040, 10979, 4077, 3114, 7191, 10305, 5554, 3917, 9471, 1446, 10917, 421, 11338, 11759, 11155, 10972, 10185, 9215, 7458, 4731, 247, 4978, 5225, 10203, 3486, 1747, 5233, 6980, 271, 7251, 7522, 2831, 10353, 1242, 11595, 895, 548, 1443, 1991, 3434, 5425, 8859, 2342, 11201, 1601, 860, 2461, 3321, 5782, 9103, 2943, 104, 3047, 3151, 6198, 9349, 3605, 1012, 4617, 5629, 10246, 3933, 2237, 6170, 8407, 2635, 11042, 1735, 835, 2570, 3405, 5975, 9380, 3413, 851, 4264, 5115, 9379, 2552, 11931, 2541, 2530, 5071, 7601, 730, 8331, 9061, 5450, 2569, 8019, 10588, 6665, 5311, 34, 5345, 5379, 10724, 4161, 2943, 7104, 10047, 5209, 3314, 8523, 11837, 8418, 8313, 4789, 1160, 5949, 7109, 1116, 8225, 9341, 5624, 3023, 8647, 11670, 8375, 8103, 4536, 697, 5233, 5930, 11163, 5151, 4372, 9523, 1953, 11476, 1487, 1021, 2508, 3529, 6037, 9566, 3661, 1285, 4946, 6231, 11177, 5466, 4701, 10167, 2926, 1151, 4077, 5228, 9305, 2591, 11896, 2545, 2499, 5044, 7543, 645, 8188, 8833, 5079, 1970, 7049, 9019, 4126, 1203, 5329, 6532, 11861, 6451, 6370, 879, 7249, 8128, 3435, 11563, 3056, 2677, 5733, 8410, 2201, 10611, 870, 11481, 409, 11890, 357, 305, 662, 967, 1629, 2596, 4225, 6821, 11046, 5925, 5029, 10954, 4041, 3053, 7094, 10147, 5299, 3504, 8803, 365, 9168, 9533, 6759, 4350, 11109, 3517, 2684, 6201, 8885, 3144, 87, 3231, 3318, 6549, 9867, 4474, 2399, 6873]

*Для Метода середины квадрата при seed = 2134:*

[0.5539, 0.6805, 0.308, 0.4864, 0.6584, 0.349, 0.1801, 0.2436, 0.934, 0.2356, 0.5507, 0.327, 0.6929, 0.011, 0.0121, 0.0146, 0.0213, 0.0453, 0.2052, 0.2107, 0.4394, 0.3072, 0.4371, 0.1056, 0.1151, 0.3248, 0.5495, 0.195, 0.8025, 0.4006, 0.048, 0.2304, 0.3084, 0.511, 0.1121, 0.2566, 0.5843, 0.1406, 0.9768, 0.4138, 0.123, 0.5129, 0.3066, 0.4003, 0.024, 0.0576, 0.3317, 0.0024, 0.0005, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

В результате мы видим недостаток работы Метода середины квадрата — если на пути встречается число 0, то все последующие обращаются в ноль.

**Вывод:** в результате работы было реализовано два генератора псевдослучайных чисел и выведены по 1000 значений при заданных начальных значениях.

**Литература**

1. Тюльпинова, Н. В. Алгоритмизация и программирование : учебное пособие / Н. В. Тюльпинова. — Саратов : Вузовское образование, 2019. — 200 c. — ISBN 978-5-4487-0470-3. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROFобразование : [сайт]. — URL: <https://profspo.ru/books/80539>
2. Соснин В.В. Облачные вычисления в образовании / Соснин В.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 109 c. — ISBN 978-5-4486-0512- 3. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/79705.html>
3. Шаманов А.П. Системы счисления и представление чисел в ЭВМ : учебное пособие / Шаманов А.П.. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2016. — 52 c. — ISBN 978-5-7996-1719-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/66204.html>
4. Минитаева А.М. Кодирование информации. Системы счисления. Основы логики : учебное пособие / Минитаева А.М.. — Москва : Московский государственный технический университет имени Н.Э. Баумана, 2019. — 108 c. — ISBN 978-5-7038-5244-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/110640.html>
5. Широков А.И. Информатика: разработка программ на языке программирования Питон: базовые языковые конструкции : учебник / Широков А.И., Пышняк М.О.. — Москва : Издательский Дом МИСиС, 2020. — 142 c. — ISBN 978-5-907226-76-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/106713.html>