Exercise 1. A simple congruential random number generator

Goal: In this exercise we are going to implement a basic congruential random number generator (proposed by Lehmer in 1948).

Write a program that generates random numbers according to

$$x_i = (cx_{i-1}) \bmod p.$$

At first, consider $c = 3$ and $p = 31$.

Task 1: Check your generated random numbers for correlations using the square test. Plot x_i vs x_{i+1}.

Task 2: Repeat the same using the cube test. Plot x_i vs x_{i+1} vs x_{i+2}.

Task 3: Repeat the same for a different random number generator. This can be achieved e.g. by choosing a different c and p.

Task 4: **[Optional]** Check the built-in random number generators your programming language provides.

Exercise 2. The χ^2-test

Goal: Here, we are getting to know to another random number generator test.

Task: Test your random number generators from exercise 1 using the χ^2-test.

The χ^2-test is described in the following:

- Divide the range of random numbers into k bins i.e. discrete intervals of the same size such that the probability of a random number to be in the interval i is given by $p_i = 1/k$.
- Using each random number generator, generate at least one sequence of n numbers. For each sequence, measure the count N_i of random numbers in each interval i.
- Compute the χ^2-value for one specific sequence of random numbers

$$\chi^2 = \sum_{i=1}^{k} \frac{(N_i - np_i)^2}{np_i}.$$

- Use the table `chi_square_description.pdf` (from Donald E. Knuth, The Art of Computer Programming, Volume 2) to check the reliability of your random number generators.

Exercise 3. Random numbers on a circle

Goal: Finally, we are learning how random numbers are distributed uniformly on a circle.

Task: Consider a circle of radius R. Generate a homogeneous distribution of random points within the circle.

Hint: If we draw (r, φ) uniformly in $[0, R) \times [0, 2\pi)$, the points are not homogeneously distributed within the circle!