MA 323 (2020) Monte Carlo Simulation Lab 03

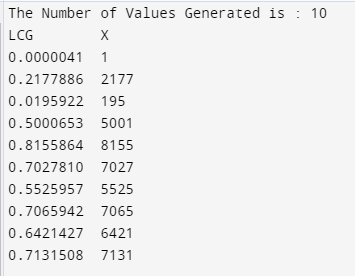
**Name:** Udandarao Sai Sandeep

**Roll Number:** 180123063

**Dept.:** Mathematics and Computing

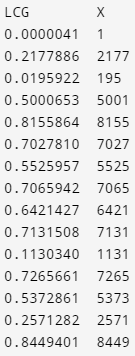
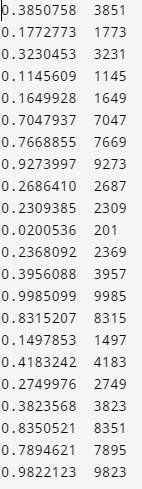
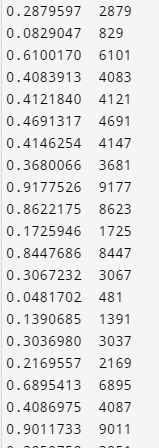
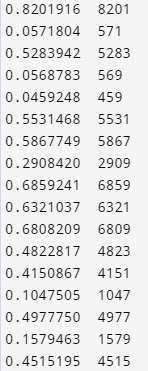
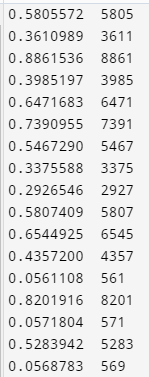
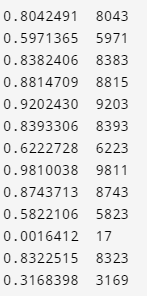
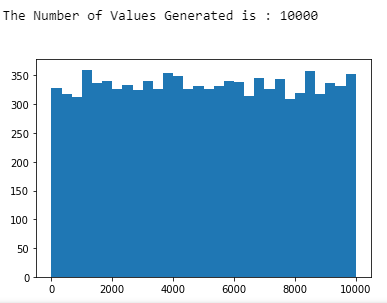
**Q1.**

**Case1**:

10 values were generated from the Discrete Distribution.

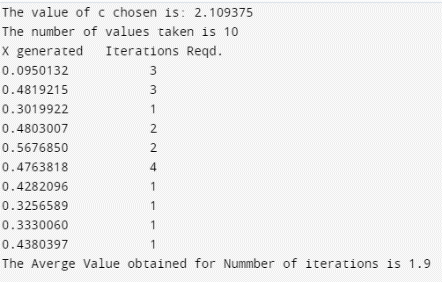
Firstly, 10 values were generated (within the range [0,1)) using Linear Congruence Generator. The parameters for the LCG are as follows: a = 1597, b = 51749, m = 244944, x0 = 1

From the given discrete distribution, the array q (Cumulative Distribute for the Discrete RV) and the array c (Probability Mass Function) were created. For each random float number (let it be denoted by **U**) generated within the range [0,1), the value of **k** was obtained such that q[k-1] < U <= q[k]. Then, c[k] was chosen to be the required X value.

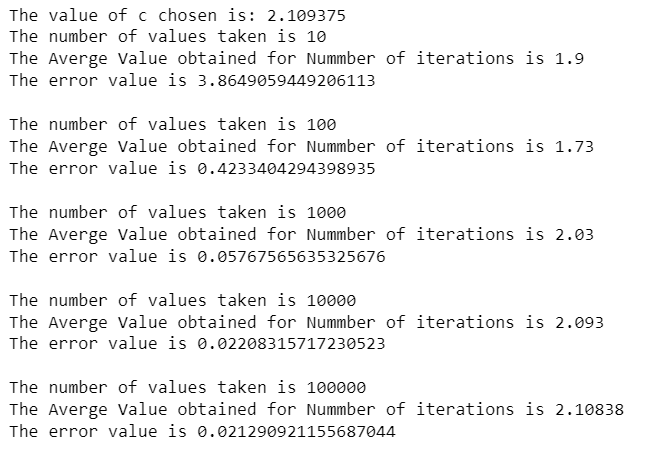
**Case2:** 100 values were generated from the Discrete Distribution.

The table above shows the generated values of X (right) and the corresponding value generated by LCG (left). Same Procedure was followed as Case-1.

**Case3:** 10000 values of X were generated. A histogram to the left depicts that the generated distribution is mimicking uniformity (i.e. the probability of each value from the discrete distribution being chosen is approximately same).

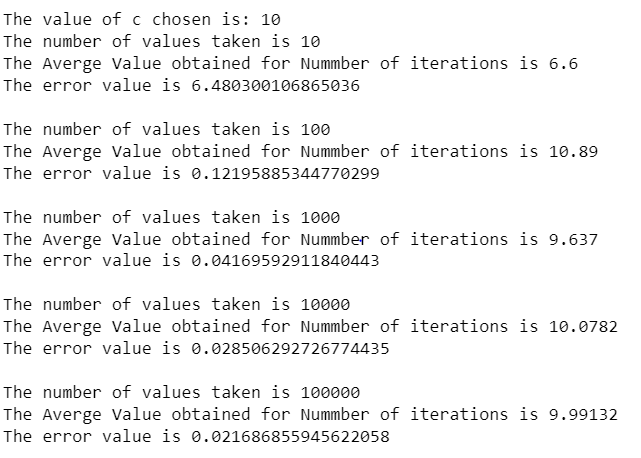
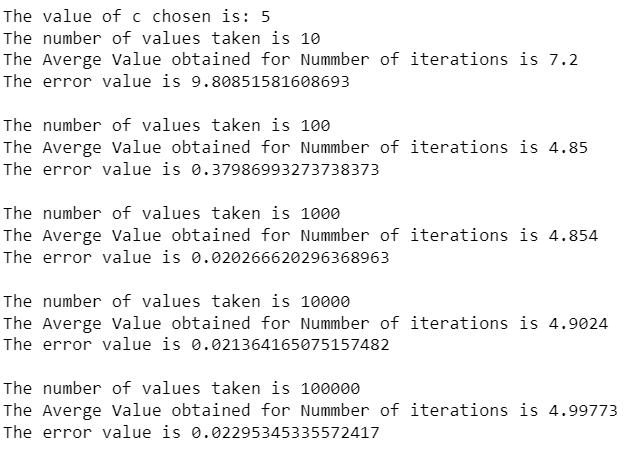
**Q2.** (a)f(x) = 20x(1-x)3. Now, g(x) denotes the probability density function of U [0,1]. So, g(x) = 1, for all x within the range [0,1]. Now, we are required to find the minimum value of c such that the inequality f(x) < cg(x) is satisfied for all x within the range [0,1]. So, we are to find the maximum value that f(x)/g(x) attains within the range [0,1]. Upon differentiating, we see that f(x)/g(x) attains its max value at x = ¼. [0 <= x <= 1]. And, the maximum value of f(x) turns out be 2.109375. So, minimum possible value of c is **2.109375**.

(b) Keeping c = 2.109375, random variables were generated from f(x) using the acceptance-rejection method. The following table to the left shows 10 generated values of f(x) (and the number of iterations required in each case). Using python inbuilt distribution builder, a set of coordinates (xi, yi) were obtained using the generated X values (similar procedure followed as the Frequency histograms in the previous assignments).

Using the coordinates and the given function f(x), the error was calculated (in a similar fashion to regression) using the formula: **error** = . As shown in the screenshots, the error value diminishes to 0 as the number of values generated increases. Hence, the distribution formed by the sample X values converges to the distribution of f(x).

(c) The experimented was repeated for different number of total values generated. The outcomes can be seen in the screenshots. It can be seen that as the number of generated values increases, the avg. number of iterations converges to c. The outcome of this experiment signifies that the expected number of iterations required for a generated value to be accepted is c. This is true, because, it can be proved that the Acceptance probability (Probability that a certain value is accepted) is 1/c. (So, the expected value of the iterations required is c).

(d) The above experiment was repeated with 2 higher values of c (5 and 10). In both these cases, the average number of iterations converge to their respective values of c (supporting the fact that the acceptance probability is 1/c as mentioned above). Since, it takes longer times to generate the values of X when values of are higher, it is always more practical to ensure that value of c is closer to the minimum possible value. Larger values of c decrease the acceptance probability, and hence lead to larger run times. Also, as number of generated values increases, less the is the error between the sample distribution and f(x).



**Q3.**

The values of c chosen for are: c = 2, and c = 3. These values are possible (all values of c >= 1.2 are possible). For each value of c, the experiment is conducted for 10,100 and 1000 values (number of values generated using acceptance rejection method).

**Note:** During performing acceptance and rejection method, Random values for U [0,1] have been obtained though random module of python. For generating the values of distribution **g**, the technique for generating values for Discrete distributions was employed.

After generating the values, a frequency table was also generated.

The first observation is that higher values of c implies higher Avg. Acceptance Iterations.

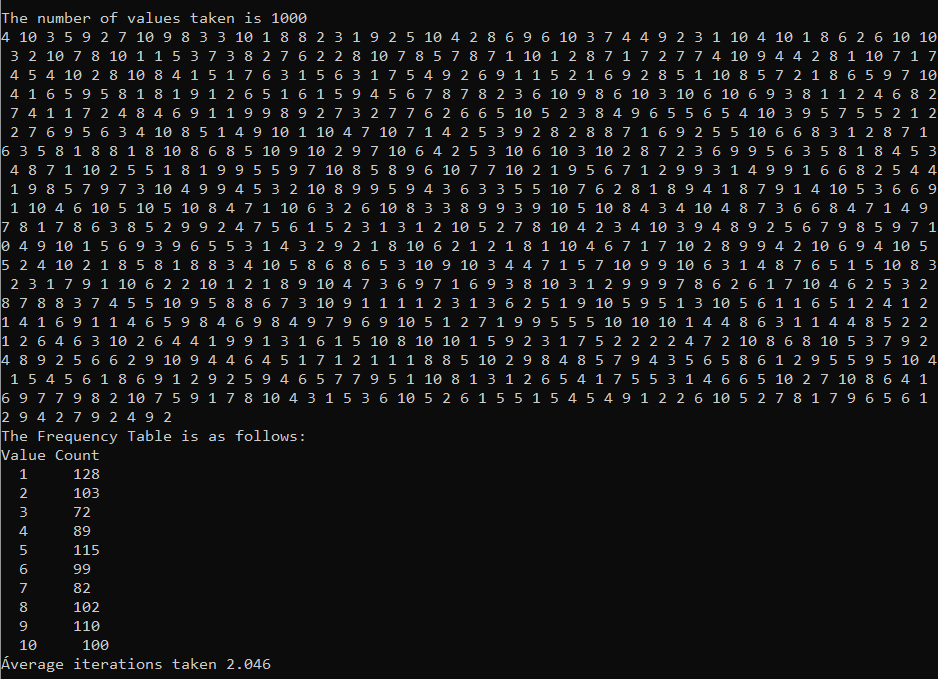
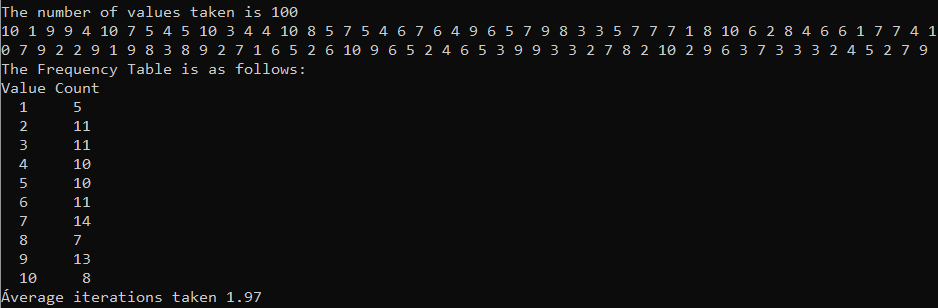
The second observation is that as the number of generated values increases, the avg. value of acceptance iterations converges to the value of c.

Also, from the frequency tables, as the number of generated values reaches larger values, the generated values seem to follow the distribution followed by f(x). (i.e. (frequency of certain value)/ (number of values) is approximately equal to f (certain value) where f is the probability mass function of X).

The outcome of the experiments is as follows:

**The value of c is 2**





**The value of c is 3**



