Experiment No.: 3

Title: Implementation of Uniformity test

Batch: B–1 **Roll No.: 16010422234** **Experiment No.: 3**

Aim: To implement Kolmogorov –Smirnov (K S) test / Chi-square test on the random number generator implemented in experiment no 1 for uniformity testing.

Resources needed: Turbo C / Java / python

Theory

Problem Statement:
Write a function in C / C++ / java / python or macros in MS-excel to implement Kolmogorov-Smirnov (KS) / Chi-square test.

Concepts:
Random Numbers generated using a known process or algorithm is called Pseudo random Number. The random numbers generates must possess the property of :

1. Uniformity
2. Independence

Uniformity:
If the interval (0, 1) is divided into "n" classes or subintervals of equal length, the expected number of observations in each interval is N/n, where N is the total number of observations.

Tests for Random numbers

1) Uniformity Test
A basic test that is to be performed to validate a new generator is the test of uniformity.
Two different testing methods are available, they are

1. Kolmogorov- Smirnov Test
2. Chi-square Test

Both of these measure the degree of agreement between distance of sample of generated random numbers and the theoretical uniform distributions.

a) Kolmogorov–Smirnov Test: This test compares the continuous cdf F(x) of the uniform distribution to the empirical cdf $S_N(x)$ of sample of N distribution
By definition,
$F(x) = x \qquad 0 \leq x \leq 1$

If the sample from random no. generated is $R_1, R_2, \ldots, R_N$ then the empirical cdf $S_N(x)$ is defined as

$S_N(x)$ = No. of $R_1, R_2, \ldots, R_N$ which are x N

As N becomes larger $S_N(x)$ should become a better approximation to F(x) provided the null hypothesis is true. The Kolmogorov-Smirnov distance test is best on the largest absolute deviation between F(x) & $S_N(x)$ over a range of random variables.

2) Chi square test: The Chi square test sample test statistics is: $X_0^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$

Where, O_i = Observed frequency in ith class, E_i = Expected frequency in ith class, n = is the no. of classes

---

Procedure: ***(Write the algorithm for the test to be implemented and follow the steps given below)***

Steps:

- Make a null hypothesis for uniformity
- Generate 5 sample sets (Each set consisting of 100 random numbers) of Pseudo random numbers using Linear Congruential Method implemented in expt 1
- Implement either Kolmogorov-Smirnov Test or Chi-square Test
- Execute the test using all the five sample sets of random numbers as input and using α=0.05.
- Draw conclusions on the acceptance or rejection of the null hypothesis of uniformity.

Results: (Program printout with output)

```
import numpy as np
from scipy.stats import chisquare, chi2
import math

def linear_congruential_generator(seed, a, c, m, n):
    x = seed
    random_numbers = []
    for _ in range(n):
        x = (a * x + c) % m
        random_numbers.append(x / m)
    return random_numbers

def chi_square_test(random_numbers, num_classes, alpha):
    observed_freq, bin_edges = np.histogram(random_numbers,
bins=num_classes, range=(0, 1))
    expected_freq = [len(random_numbers) / num_classes] * num_classes

    chi_stat, p_value = chisquare(observed_freq, expected_freq)

    df = num_classes - 1
```

```
    critical_value = chi2.ppf(1 - alpha, df)

    if chi_stat > critical_value:
        return chi_stat, p_value, critical_value, False
    else:
        return chi_stat, p_value, critical_value, True

def main():
    seed = int(input("Enter the seed (X0): "))
    a = int(input("Enter the multiplier (a): "))
    c = int(input("Enter the increment (c): "))
    m = int(input("Enter the modulus (m): "))
    n = int(input("Enter the number of random numbers to generate (n): 
"))

    random_numbers = linear_congruential_generator(seed, a, c, m, n)

    print("\nGenerated Random Numbers (Scaled to [0, 1]):")
    print(random_numbers)

    num_classes = math.ceil(1 + math.log2(n))
    print(f"\nAutomatically calculated number of classes (bins): 
{num_classes}")

    alpha = float(input("Enter the significance level (alpha, typically 
0.05): "))

    chi_stat, chi_p_value, critical_value, accept_null = 
chi_square_test(random_numbers, num_classes, alpha)

    print(f"\nChi-Square test statistic: {chi_stat}")
    print(f"Standard-value: {chi_p_value}")
    print(f"Critical value (Chi-Square table value for df={num_classes-1} 
and alpha={alpha}): {critical_value}")

    if accept_null:
        print("Chi-Square test: Accept the null hypothesis (uniform 
distribution).")
    else:
        print("Chi-Square test: Reject the null hypothesis (not 
uniform).")

if __name__ == "__main__":
    main()
```

Test Case 1: Uniform Distribution (Accept Null Hypothesis)

- Seed (X0): 5
- Multiplier (a): 1103515245
- Increment (c): 12345
- Modulus (m): 2147483648
- Number of Random Numbers (n): 1000
- Significance Level (alpha): 0.05

Expected Result: The generated random numbers should approximate a uniform distribution, so the null hypothesis will likely be accepted.

```
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/Downloads/VI SEM/MS/EXP03/EXP03.py"
Enter the seed (X0): 5
Enter the multiplier (a): 1103515245
Enter the increment (c): 12345
Enter the modulus (m): 2147483648
Enter the number of random numbers to generate (n): 1000

Generated Random Numbers (Scaled to [0, 1]):
[0.5693273963406682, 0.2580903223715726, 0.3240002878010273, 0.9728268985636532, 0.3110656915232539, 0.7923837187699974, 0.5524904821068048, 0.7222646162845194, 0
.9940482014790177, 0.5969333662651479, 0.9227651953697205, 0.6458956864662468, 0.6952493181452155, 0.6491061891429126, 0.3430633153766397, 0.51837020600214 6, 0.87
71643685176969, 0.030082283075898886, 0.9786656610667706, 0.7451900173909962, 0.6127852005884051, 0.7596937823109329, 0.3118315059691667 6, 0.7082897634245455, 0.81
64351442828774, 0.2699356018565595, 0.8169694766402245, 0.17216481687501073, 0.07421334739774466, 0.23589805467054248, 0.5947928261011839, 0.21929608518257737, 0.1
677984865382313 7, 0.9828713438473642, 0.809209082275629, 0.6836216901428998, 0.885361853055656, 0.1883719298057258, 0.27069407142698765, 0.5508055225946009, 0.2133
3981398493052, 0.0978407724760472 8, 0.009900316596031189, 0.29405267210677266, 0.5028156479820609, 0.9727634484879673, 0.1852498110383749, 0.614221731666475 5, 0.70
42607562616467, 0.9899621433578432, 0.16826116666197777, 0.5529839773662388, 0.2643852485343814, 0.3108095186762512, 0.15036113746464252, 0.4477794230915606, 0.778
847903944552, 0.5391144542954862, 0.11493051797151566, 0.6973197502084076, 0.9945755070075393, 0.28642966551706195, 0.5183344166725874, 0.8063881122507155, 0.25544
1558547318, 0.063531195279210 8, 0.5236869044601917, 0.6786858146078885, 0.9850543616339564, 0.21681978134438396, 0.13110003806650639, 0.626475869212299 6, 0.3004044
8252111673, 0.12839409848675132, 0.048167258501052856, 0.06577342422679067, 0.35012158658355474, 0.39854587567970157, 0.6444311644881964, 0.36583306873217225, 0.47
10906492546201, 0.22942688828334212, 0.8335856609046459, 0.8216830124147236, 0.7571775419637561, 0.7286378252319992, 0.22716248221695423, 0.21845614025369287, 0.13
381399307399988, 0.3514890312217176, 0.4034520909190178, 0.9562679422087967, 0.5321919238194823, 0.2006831238977611, 0.6354089509695768, 0.204391317907720 8, 0.2568
171592429280 3, 0.4021694860421121, 0.9212911762297153, 0.05347825633361935 6, 0.14017251413315535, 0.2759206327609718, 0.6617845539003611, 0.13457839051261544, 0.57
82402539625764, 0.52038045367226, 0.8273609429597855, 0.6737044272013009, 0.04063393082469702, 0.12933433009311557, 0.9596210476011038, 0.45069445157423615, 0.1490
8959437161684, 0.2599511272273958, 0.85037156566977 5, 0.6311211069114506, 0.9180663777515292, 0.7707470613531768, 0.24218671210110188, 0.9399976492859423, 0.251206
45854622126, 0.648221442010253 7, 0.39420411735773087, 0.14603087725117803, 0.2874043947085738, 0.04091428918763995, 0.8569051008671522, 0.3251709365285933, 0.69023
58355000615, 0.1196358255110681, 0.2996293194591999, 0.8722079996950924, 0.4744956074282527, 0.4826178434304893, 0.7345737908035517, 0.729165820870548 5, 0.46359517
51843095, 0.3243369474075734 6, 0.981026291847229, 0.7992421654053032, 0.9715694775804877, 0.08675967110320926, 0.7135831322520971, 0.015046114567667246, 0.80344313
8487637, 0.811759469565004 1, 0.93810130283236 5, 0.0298822452314198, 0.1677060527727008, 0.9134555864147842, 0.23913499899208546, 0.0008316845633089542, 0.594648347
2362161, 0.5892237960360944, 0.642606519162654 9, 0.4323800397105515, 0.4543047165498137 5, 0.5881290207616985, 0.43746153824031353, 0.54934220248833 3, 0.16775813419
371843, 0.555529823061078 8, 0.8000587560236454, 0.1678340290673077, 0.7055529383942485, 0.6726047727279365, 0.5650439206510782, 0.5330408946610987, 0.4669672930613
16, 0.8095506890676916, 0.9864582270383835, 0.09253312321379781, 0.13389502372592688, 0.9112027580849826, 0.8328310903161764, 0.6738782920874655, 0.593086828477680
7, 0.8338265246711671, 0.66000729426741 6, 0.035300411749631166, 0.5205008639022708, 0.35183175979182124, 0.605458514764905, 0.258135980460792 8, 0.721512706018984 3,
 0.5531582026742399, 0.547829218208789 8, 0.9498369139619172, 0.8207346862182021, 0.3420831779949367, 0.9754669349640608, 0.7262703501619399, 0.395194602198898 8, 0.
2682010880671441 6, 0.40768690779805183, 0.9420653288252652, 0.1446238039061427, 0.4003247800283134, 0.7125211153179407, 0.13775634625926614, 0.19260465446859598, 0
```

```
842, 0.2792460094206035, 0.5010553430765867, 0.6737244003452361, 0.7094570072367787, 0.15786640858277678, 0.5444987751543522, 0.266606162674725, 0.29225665610283 6
13, 0.462207708042115, 0.1809887234121561, 0.45840842882171273, 0.6412631357088685, 0.311246023979038, 0.40650974214076996, 0.6933643338270485, 0.7174230059608817,
 0.1915645892731845 4, 0.665128355845809, 0.5576408305205405, 0.7138834493234754, 0.481645745690912, 0.05932020768523216, 0.517225601244717 8, 0.07784285675734282, 0
.14608481107279658, 0.08178158290684223, 0.49793756613507867, 0.2882607886567712, 0.8184758215211332, 0.7124753072857857, 0.27592981280758977, 0.9831773107871413,
0.9917191485874355, 0.22466102056205273, 0.14748940104618669, 0.5303917052224278, 0.5345009914599359, 0.5436598248779774, 0.8468840424902737, 0.6352505562826991, 0
.2526947441510856, 0.5021033082157373, 0.1810056553222239, 0.07929521147161722, 0.7144342376850545, 0.8354169428348541, 0.34956079395487905, 0.18351861741393805, 0
.05760885728523135, 0.26128785870969296, 0.4195579537190497, 0.08998153079301119, 0.9985305345617235, 0.4868669919669628, 0.9228417505510151, 0.4555381042882 8, 0.7
605226146988571, 0.48745560832321644, 0.04542397568002343, 0.6514208475127816, 0.14118059864267707, 0.9004262015223503, 0.37736152531579137, 0.06243496295064688, 0
.43705476028844714, 0.8781277667731047, 0.691931205336004 5, 0.5795120568946004, 0.4445046386681497, 0.2435254491865635, 0.7228514109738171, 0.8793732253834605, 0.2
5547540513798594, 0.2923245672136545, 0.40830068150535226, 0.5850515151396394, 0.06694610556587577, 0.08532901108264923, 0.5704831299372017, 0.9010237595066428, 0.
7227997775189579, 0.5747840162366629, 0.4994907476939261, 0.8167017893865705, 0.20686544245108962, 0.4084533117711544, 0.41021258430555463, 0.47203301917761564, 0.
8058819728903472, 0.7551806438714266, 0.2410408020950853 8, 0.7789604077115655, 0.16113383462652564, 0.9956856742501259, 0.7631236058659852, 0.8924917941913009, 0.9
275087038986385, 0.6223442535847425, 0.4689150475896895, 0.6251286091282964, 0.2587269595824182, 0.19170306995511055, 0.2087717060931027, 0.3984039602801204, 0.837
493050377816, 0.6734786722809076, 0.0443462417460978, 0.825280093587935, 0.6693187369965017, 0.03979086875915527, 0.2875278261490166, 0.517155229113996, 0.35876820
934936404, 0.4383805003017187, 0.1936794468201697, 0.20922977197915316, 0.08687507873401046, 0.7935615889728069, 0.2779220868833363, 0.7979818983003497, 0.00848124
5022267103, 0.17865786142647266, 0.7232157797552645, 0.3845025161281228, 0.2882476388476789, 0.8036736473441124, 0.8489875388331711, 0.9174396013841033, 0.49408683
87930095, 0.9619491528719664, 0.10905616218224168, 0.5293019087985158, 0.5667675635777414, 0.779550109058618 5, 0.5876039122231305, 0.15987205784767866, 0.084441039
24557567, 0.11114179529249668, 0.46194507041946054, 0.560478993691504, 0.04083924135193229, 0.42609743773937225, 0.4008413669653237, 0.2728798231109977, 0.85589501
21514499, 0.028590982779860497, 0.36711428547278047, 0.6765010254457593, 0.837534052785486, 0.4554242454469204, 0.7933042426593602, 0.6977890385314822, 0.813388790
4696167, 0.3953384328633547, 0.5991266430355608, 0.27542021963745356, 0.1511841206813685, 0.9735600426793098, 0.01947480859234929, 0.17512018140405416, 0.88654502
43465602, 0.7453311365097761, 0.711719783488661, 0.24784250278025866, 0.17697606002911925, 0.24217398837208748, 0.11105701373890042, 0.7250568149611354, 0.80076273
47821307182, 0.6278568995185196, 0.29712533112615347, 0.5733891199342906, 0.1646288763731718, 0.8450211458839476, 0.830310937948525, 0.11645205831155181, 0.6584321
297705173, 0.9995899950154126, 0.24898752849549055, 0.5096514853648841, 0.7370497826486826, 0.47676347056403756]

Automatically calculated number of classes (bins): 11
Enter the significance level (alpha, typically 0.05): 0.05

Chi-Square test statistic: 8.941999999999999
Standard-value: 0.5376167218755842
Critical value (Chi-Square table value for df=10 and alpha=0.05): 18.307038053275146
Chi-Square test: Accept the null hypothesis (uniform distribution).
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03> 
```

Test Case 2: Non-Uniform Distribution (Reject Null Hypothesis)

- Seed (X0): 7
- Multiplier (a): 5
- Increment (c): 3
- Modulus (m): 16
- Number of Random Numbers (n): 100
- Significance Level (alpha): 0.01

Expected Result: With a small modulus, the generated numbers will exhibit poor randomness and fail the Chi-Square test, rejecting the null hypothesis.

```
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/Downloads/VI SEM/MS/EXP03/EXP03.py"
Enter the seed (X0): 7
Enter the multiplier (a): 5
Enter the increment (c): 3
Enter the modulus (m): 16
Enter the number of random numbers to generate (n): 100

Generated Random Numbers (Scaled to [0, 1]):
[0.375, 0.0625, 0.5, 0.6875, 0.625, 0.3125, 0.75, 0.9375, 0.875, 0.5625, 0.0, 0.1875, 0.125, 0.8125, 0.25, 0.4375, 0.375, 0.0625, 0.5, 0.6875, 0.625, 0.3125, 0.75,
 0.9375, 0.875, 0.5625, 0.0, 0.1875, 0.125, 0.8125, 0.25, 0.4375, 0.375, 0.0625, 0.5, 0.6875, 0.625, 0.3125, 0.75, 0.9375, 0.875, 0.5625, 0.0, 0.1875, 0.125, 0.812
5, 0.25, 0.4375, 0.375, 0.0625, 0.5, 0.6875, 0.625, 0.3125, 0.75, 0.9375, 0.875, 0.5625, 0.0, 0.1875, 0.125, 0.8125, 0.25, 0.4375, 0.375, 0.0625, 0.5, 0.6875, 0.62
5, 0.3125, 0.75, 0.9375, 0.875, 0.5625, 0.0, 0.1875, 0.125, 0.8125, 0.25, 0.4375, 0.375, 0.0625, 0.5, 0.6875, 0.625, 0.3125, 0.75, 0.9375, 0.875, 0.5625, 0.0, 0.18
75, 0.125, 0.8125, 0.25, 0.4375, 0.375, 0.0625, 0.5, 0.6875]

Automatically calculated number of classes (bins): 8
Enter the significance level (alpha, typically 0.05): 0.01

Chi-Square test statistic: 0.16
Standard-value: 0.9999882998398463
Critical value (Chi-Square table value for df=7 and alpha=0.01): 18.475306906582357
Chi-Square test: Accept the null hypothesis (uniform distribution).
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03>
```

Test Case 3: Uniform Distribution (Accept Null Hypothesis)

- Seed (X0): 42
- Multiplier (a): 6364136223846793005
- Increment (c): 1442695040888963407
- Modulus (m): 18446744073709551616
- Number of Random Numbers (n): 5000
- Significance Level (alpha): 0.10

Expected Result: This configuration generates high-quality random numbers with a large period, likely passing the Chi-Square test.

```
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/Downloads/VI SEM/MS/EXP03/EXP03.py"
Enter the seed (X0): 42
Enter the multiplier (a): 6364136223846793005
Enter the increment (c): 1442695040888963407
Enter the modulus (m): 18446744073709551616
Enter the number of random numbers to generate (n): 5000
```

```
8996477945816, 0.5246903178358113, 0.9735180431513395, 0.106443769405765, 0.2077616339260307, 0.9102906411553735, 0.11747880537127783, 0.5931591742313448, 0.397006
9355100749, 0.9224309212029719, 0.5250590615377213, 0.5013891492267898, 0.6174059578713406, 0.0002667702189270723, 0.3881318325899969, 0.16743408845161606, 0.16884
958908728567, 0.13316554432072422, 0.23176787629186738, 0.3653627191047432, 0.5409286038584545, 0.49825236353287167, 0.4226697897303697, 0.19917366894869046, 0.869
14431769325, 0.526715113521946‌8, 0.6292662826868282, 0.061297069685725604, 0.5207054974485845, 0.38716591388469107, 0.635635298679233, 0.9538379253401632, 0.952503
3405141297, 0.5210912957097806, 0.15260594402411712, 0.4748060637213526, 0.6986478859855959, 0.6457225099824074, 0.060991070667242596, 0.07834033528686261, 0.79598
8159944879, 0.4750475543728656, 0.5951592323491226, 0.9604547310884235, 0.9558160556796556, 0.3327498377578312, 0.6520042314446274, 0.6512506295713044, 0.772184515
0110812, 0.5383676089548353, 0.26698573714434237, 0.9479875422349707, 0.14650831214408516, 0.6866020834915298, 0.15473687754674015, 0.513106052549854‌8, 0.796611378
1977844, 0.76438046358824, 0.9957649792840094, 0.1621483875554482, 0.9597506164208148, 0.1357825495646276, 0.5250018960754362, 0.3469115597973265, 0.38152646892657
427, 0.002097665939743201‌5, 0.6759553218460032, 0.19536369904734083, 0.24715489502996071, 0.572653320865181, 0.2009721801966697, 0.28994983112263‌6, 0.1352839987329
2352, 0.46164365358491816, 0.585487415443416, 0.26492709932527186, 0.5392464556087234, 0.29130788936635504, 0.8440965540246133, 0.7367241557315951, 0.8237367728838
506, 0.31099181991383595, 0.542775922504080‌7, 0.8595561042058056, 0.38981275352182526, 0.6578616477515122, 0.7392333557983983, 0.491726680311979, 0.97909529312393‌0
2, 0.8114884412894398, 0.6902315575870797, 0.39406962241566174, 0.32583048993789665, 0.07130277303918355, 0.2035292455795339, 0.20822372463074088, 0.16656848507852
023, 0.744293290516002‌9, 0.24334514082605704, 0.7565097045570248, 0.9074481050313913, 0.8611701920602832, 0.9561480876765315, 0.385423418527720‌8, 0.81061286645954‌0
2, 0.08335233835133628, 0.20396252662535555, 0.8772471395751245, 0.2219524941955667, 0.6916397877124139, 0.9502545494585266, 0.508490385878463‌5, 0.4006245397440371
, 0.49074541579040964, 0.6854627337519157, 0.38493255815432664, 0.13962745969663296, 0.8891092847632567, 0.3847263542220798, 0.77398121334489‌3, 0.15277790281195244
, 0.9399843573862317, 0.4629278161588140‌6, 0.5981916875086905, 0.48141616057015, 0.7137566444484033, 0.5113650796451559, 0.485933643950648‌2, 0.9683693967697414, 0.
6803275334327753, 0.7417610826905943, 0.20709385097035893, 0.516458691515963, 0.9197873773101387, 0.322890314669848, 0.5576745083719236, 0.6260783317336238, 0.9247
906511409837, 0.6323769609514742, 0.07641978990333714, 0.49777086509521196, 0.9830898140041773, 0.08838834222102913, 0.7337460622387532, 0.264440634833069‌3, 0.3173
753879690209, 0.05358052501040167, 0.8504110073202233, 0.23332215493005748, 0.522241695762756, 0.3665819250621072, 0.49284937885987523, 0.6049174425044549, 0.44957
42212485003, 0.14645454346709855, 0.06343522594436722, 0.3083243513615292, 0.40788988269636567, 0.5516400156317284, 0.17498552803069262, 0.0566674099094869, 0.4288
9742565599337, 0.23182658515598797, 0.04655788337878849, 0.6605514618911533, 0.2508917332443037, 0.35563822246694016, 0.5092412586224557, 0.32412943891577184, 0.30
807559364025394, 0.2681456321155518, 0.2919690498050698, 0.6621233397255516, 0.3721217671977388, 0.3347557366192965, 0.4803236323957068, 0.5747376436506578, 0.3370
2544460685835, 0.2840268373445351, 0.528699476779396, 0.7899269341178474, 0.9544213544196346, 0.8936803983809793, 0.7172373970962956, 0.1609105965669666, 0.1347986
587099078, 0.06084222849880632]

Automatically calculated number of classes (bins): 14
Enter the significance level (alpha, typically 0.05): 0.10

Chi-Square test statistic: 10.263999999999998
Standard-value: 0.6722167646536351
Critical value (Chi-Square table value for df=13 and alpha=0.1): 19.81192930712756
Chi-Square test: Accept the null hypothesis (uniform distribution).
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03>
```

Test Case 4: Non-Uniform Distribution (Reject Null Hypothesis)

- Seed (X0): 1
- Multiplier (a): 2
- Increment (c): 0
- Modulus (m): 8
- Number of Random Numbers (n): 50
- Significance Level (alpha): 0.05

Expected Result: With such a poor choice of parameters (small modulus and no increment), the distribution will be non-uniform, and the null hypothesis will likely be rejected.

```
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/Downloads/VI SEM/MS/EXP03/EXP03.py"
Enter the seed (X0): 1
Enter the multiplier (a): 2
Enter the increment (c): 0
Enter the modulus (m): 8
Enter the number of random numbers to generate (n): 50

Generated Random Numbers (Scaled to [0, 1]):
[0.25, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0
.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

Automatically calculated number of classes (bins): 7
Enter the significance level (alpha, typically 0.05): 0.05

Chi-Square test statistic: 272.84000000000003
Standard-value: 5.353522947600425e-56
Critical value (Chi-Square table value for df=6 and alpha=0.05): 12.591587243743977
Chi-Square test: Reject the null hypothesis (not uniform).
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03>
```

Test Case 5: Uniform Distribution (Borderline Case)

- Seed (X0): 123
- Multiplier (a): 1664525
- Increment (c): 1013904223
- Modulus (m): 4294967296
- Number of Random Numbers (n): 2000
- Significance Level (alpha): 0.05

Expected Result: This is a standard LCG configuration for generating random numbers. Depending on the number of bins and the generated sequence, the result could borderline pass or fail the Chi-Square test.

```
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03> & C:/Users/Dell/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Dell/Downloads/VI SEM/MS/EXP03/EXP03.py"
Enter the seed (X0): 123
Enter the multiplier (a): 1664525
Enter the increment (c): 1013904223
Enter the modulus (m): 4294967296
Enter the number of random numbers to generate (n): 2000

Generated Random Numbers (Scaled to [0, 1]):
[0.2837369213812053, 0.4351300236303359, 0.03865125775337219, 0.22087990469299257, 0.3594270762987435, 0.5902441388461739, 0.361280900426209, 0.3268499083351344, 0
.07973951241001487, 0.6479622528422624, 0.6049802396446466, 0.9694624783005565, 0.7677612067200243, 0.9586836213711649, 0.0909308111295104, 0.844463321147486, 0.5
457009929232299, 0.1813135121483558, 0.10987668111920357, 0.7187079151626676, 0.5285541121847928, 0.7696523650083691, 0.8389335284009576, 0.0674295767676852, 0.4
523370764218271, 0.6081990145612508, 0.700780538842082, 0.9624840894248337, 0.06501784408465028, 0.0629929753486067, 0.6183601124212146, 0.1021958950441330, 0.858
2663084380329, 0.963120789499534, 0.8682103827595711, 0.12343084788881242, 0.9681500983424485, 0.2785114368889481, 0.4855555491521954, 0.586520530981943, 0.32290
069153532386, 0.5096458077896386, 0.4242790210992098, 0.27366313501261175, 0.3658748404122889, 0.05480523803271353, 0.9249043753370643, 0.6914258997421712, 0.93183
63103084266, 0.07048410666175187, 0.7937091253697872, 0.9179741179104894, 0.10468293027952313, 0.5905914960894734, 0.5460963035002351, 0.1856517016422003, 0.13474
39563833177, 0.9200669147539884, 0.6173488553613424, 0.8395383113529533, 0.7437727474607527, 0.5685350822750479, 0.09389184694737196, 0.562608047155634, 0.3957597
534172237, 0.239624777110293, 0.6681874841451645, 0.008114702766761184, 0.3616908160382924, 0.6415767467115074, 0.770387944765389, 0.2298285819124430, 0.6563757
820986211, 0.13476568018086255, 0.07987102307379246, 0.5507498772349209, 0.17547242948785424, 0.4817612434271723, 0.8697835868224502, 0.7609236116986722, 0.6108307
051472366, 0.2155531768221408, 0.8877178467810154, 0.7849811424966902, 0.9722822760231793, 0.3915654553566128, 0.7256454387679696, 0.2100332274567335, 0.794000392
3140466, 0.7390845112968236, 0.8822343181818724, 0.3145396539475769, 0.35355506325140595, 0.47772651934064925, 0.9706734670326114, 0.4887804302852601, 0.4817885453
812778, 0.314568744273856, 0.7751304134726524, 0.18755353963933885, 0.7916361433453858, 0.38756995112635195, 0.10896656382828951, 0.805724256439134, 0.4040173231
624067, 0.170904877828434, 0.6778353471308947, 0.11725102527998388, 0.49892213800922036, 0.6078377703670412, 0.4007881721481681, 0.168312902329489, 0.26981796650
215983, 0.9867599804420024, 0.892513196915388, 0.7651635592337698, 0.10950156347826123, 0.326016625622287, 0.05983191076666117, 0.947336849523708, 0.1055214232
651381, 0.283147312002256, 0.015578528866171837, 0.08682893752120435, 0.17329545551910996, 0.8541659193579108, 0.762987199239432, 0.503881989745423, 0.405048973
3926952, 0.3785044488031417, 0.3537120223045349, 0.7399944288190454, 0.4626979944296181, 0.6152459329459816, 0.4726048829033971, 0.8787827498745173, 0.09280285378
18052, 0.906271408544853, 0.6523760948330164, 0.5553198994603008, 0.09171713003888726, 0.6919459516648203, 0.57126285787671, 0.544575216947123, 0.29905688343569
636, 0.894968775333836, 0.13683052733540535, 0.0695809384342283, 0.44761520670726895, 0.9380123896989971, 0.30903169605880976, 0.2199502631556242, 0.947847088333
2193, 0.41077582468278706, 0.8656480889767408, 0.12137198238633573, 0.9350495883263648, 0.6520769151393324, 0.5632402701303363, 0.746706670848652, 0.157362326513
7586, 0.7626086485106498, 0.3967301622033119, 0.5093094406183809, 0.5327132833190262, 0.8139845749828964, 0.9107413785532117, 0.02920425753109157, 0.4528349130414
263, 0.2696982801426202, 0.7658223677426577, 0.7127348200883716, 0.16247556963935494, 0.8836219201330096, 0.01267737057060003, 0.03631700086407363, 0.79193124501
03462, 0.5916688193101436, 0.7775301896035671, 0.6749178504105657, 0.8710226197727025, 0.16224513039924204, 0.3117457712069154, 0.865886163664981, 0.4026424265466
6305, 0.621115557150915, 0.6088346000760794, 0.648759608855471, 0.8239981257356703, 0.7163081394974142, 0.04196490626782179, 0.871673418907448, 0.43367489380761
8, 0.938683101208880, 0.7251076847314835, 0.10499564534984529, 0.11264392407611012, 0.8637907600495964, 0.5509395273402333, 0.852813974721357, 0.4173410399816930
```

```
, 0.9279943308793008, 0.9996748410630971, 0.0008885245770215988, 0.20743953972123563, 0.5359224625863135, 0.5731044562999159, 0.9311905903741717, 0.253510541049763
56, 0.8694088305346668, 0.9697136890608817, 0.9143520370125771, 0.06047633267007768, 0.6037056338973343, 0.3563309332821518, 0.982789446599781, 0.839669474167749
, 0.0675570457242429, 0.62760211830027, 0.652031736448407, 0.3621797577943653, 0.4974106387235224, 0.6794892440084368, 0.0699511161074042, 0.6176066498737782,
0.944949123542755, 0.6759329785127193, 0.57712685689330, 0.31753829470835626, 0.6660673995502293, 0.07430431828834116, 0.63146687392145, 0.634382080985233, 0.0
694199181161820, 0.4252703108359128, 0.30021212063729763, 0.816171770682558, 0.5526633583940566, 0.21269883983768523, 0.7724487958475947, 0.567976190475747, 0.80
45196156017482, 0.2492274728137999, 0.5952533632516861, 0.34053448564372957, 0.40078410180285573, 0.3931213712785393, 0.5865953834727407, 0.916742936475202, 0.77
24043591879308, 0.602045263396576, 0.6281231585890055, 0.9366183371748775, 0.8737539858557284, 0.5893744791392237, 0.7909571891650558, 0.2513629372697323, 0.12922
687409445643, 0.5986700479406863, 0.4976164437830448, 0.2471559455152601, 0.4862767611630261, 0.06194285885430872, 0.6732024410739541, 0.529296596301719, 0.65302
70925723016, 0.15733188320882618, 0.08896614424884319, 0.6073237785603851, 0.8485761978663504, 0.5318214597646147, 0.3513826681300998, 0.471737227169796, 0.644122
599957101047, 0.4554273348767310, 0.42065366357564926, 0.7754312304314226, 0.9049018365330994, 0.965523230144754, 0.7907246695831418, 0.216705851955339, 0.544293
9340136945, 0.09658211772330105, 0.5855713505297899, 0.3883085714187473, 0.5609087632037699, 0.8951397279743105, 0.6917744120582938, 0.03929930436424911, 0.9106648
745946586, 0.6864526469726115, 0.8282700590789318, 0.4561563318129629, 0.8542789448983967, 0.8968249766621739, 0.8303465778008103, 0.873481866670772, 0.6401881403
289735, 0.40034905751235783, 0.251023730263114, 0.5106841826345772, 0.8251677923835814, 0.6556852536741644, 0.7329399613663554, 0.125261305598542, 0.8107693861238
658, 0.1485058006364852, 0.853872418403625, 0.22331126756034791, 0.4237038609571755, 0.9052277153823525, 0.3990147830918431, 0.317893923027440, 0.6182952239178
121, 0.09365976904518902, 0.263137916103005, 0.875869327923283, 0.6291294754482806, 0.9761885220650584, 0.4357583140954375, 0.843837680993601, 0.1520238476805389
, 0.7311284218449146, 0.7724393792450428, 0.8938058277126402, 0.3814413552172482, 0.9078609629068524, 0.5053505012765527, 0.7792053266894072, 0.9824756584130228, 0
.531387909548357, 0.696208951994776, 0.4418870785739273, 0.32553123915567994, 0.12192358099855483, 0.08472959231585264, 0.7607175174634904, 0.5618238891474903, 0
.1451511990744620, 0.03570739179849624, 0.08240136480890214, 0.36782651068642735, 0.658768298337236, 0.5378577606752515, 0.925155945820733, 0.43678522994741, 0
.1709461987484246, 0.4575396943837404, 0.49586206837557256, 0.045430822763592005, 0.9763285408262163, 0.5004867305979133, 0.911316464422270, 0.27401045290753245,
 0.4851938833016902, 0.584670718759298, 0.26421079388819635, 0.7077647228725255, 0.311407363275066, 0.57742340210825, 0.9244622110854834, 0.6979750371538103, 0.1
3478641887195408, 0.599940812215209, 0.7165204936172813, 0.5107062780298293, 0.6035055743996054, 0.3522954760119319, 0.863276733784005, 0.941369793843478, 0.792165
2880031615, 0.16208143532276154, 0.8371985924895853, 0.223231699783355, 0.9811498618219048, 0.7098170788958669, 0.5093171156477183, 0.3079914911650121, 0.772899414
6455079, 0.634230786934495, 0.23669011308811605, 0.8465559692122042, 0.8107209170702845, 0.47055438812822104, 0.778967099962756]

Automatically calculated number of classes (bins): 12
Enter the significance level (alpha, typically 0.05): 0.05

Chi-Square test statistic: 7.972
Standard-value: 0.7158098197529434
Critical value (Chi-Square table value for df=11 and alpha=0.05): 19.67513757268249
Chi-Square test: Accept the null hypothesis (uniform distribution).
PS C:\Users\Dell\Downloads\VI SEM\MS\EXP03>
```

Questions:

1. List down the pros and cons of the Kolmogorov - Smirnov test and Chi- Square test.

Ans: Pros and Cons of the Kolmogorov–Smirnov (K-S) Test and Chi-Square Test:

Kolmogorov–Smirnov Test

Pros:
- Non-parametric: Does not assume specific data distribution.
- Suitable for small sample sizes.
- Directly compares empirical and theoretical distributions.
- Easy to compute with continuous distributions.

Cons:
- Sensitive to extreme differences at particular points.
- Difficult to use with discrete data or small intervals.
- Less effective for large sample sizes when small deviations may be overemphasized.

Chi-Square Test:

Pros:
- Works well with discrete data.
- Handles large sample sizes efficiently.
- Easy to implement and interpret.
- Widely used and understood.

Cons:
- Sensitive to binning (number of classes affects results).
- Requires a sufficiently large sample size to ensure expected frequencies are meaningful.
- Results may vary based on how data is grouped into intervals.

2. What is the minimum sample size to apply each of the uniformity and independence tests?

Ans: Kolmogorov–Smirnov Test: No strict minimum, but typically works well with small sample sizes (≥ 30).

Chi-Square Test: Requires each expected frequency $E\_i \geq 5$. For n bins, $N \geq 5 \times n$, where N is the total sample size.

3. Why is it essential to test the random number generator?

Ans: Importance of Testing the Random Number Generator:
- Ensures the generator produces numbers that approximate true randomness.
- Validates uniformity and independence, critical for simulation and modeling.
- Detects potential flaws in the generator's algorithm.
- Prevents biases in applications like cryptography, gaming, and statistical sampling.
- Verifies adherence to expected statistical properties of random numbers.

Outcomes: C02 – Generate pseudorandom numbers and perform empirical tests to measure the quality of a pseudorandom number generator

---

Conclusion:

The implementation of the Chi-Square test successfully evaluates the uniformity of random numbers generated using the Linear Congruential Method. The test results confirm whether the random numbers approximate a uniform distribution. In most cases with well-designed parameters (e.g., large modulus and high-quality multipliers), the null hypothesis of uniformity is accepted. For poorly chosen parameters, randomness and uniformity fail, and the hypothesis is rejected. This experiment emphasizes the importance of testing pseudorandom number generators for ensuring statistical validity in practical applications.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. "Linear Congruential Generators" by Joe Bolte, Wolfram Demonstrations Project.
2. Severance, Frank (2001). System Modeling and Simulation. John Wiley & Sons, Ltd. p. 86. ISBN 0-471-49694-4.
3. The GNU C library's rand() in stdlib.h uses a simple (single state) linear congruential generator only in case that the state is declared as 8 bytes. If the state is larger (an array), the generator becomes an additive feedback generator and the period increases. See the simplified code that reproduces the random sequence from this library.
4. "A collection of selected pseudorandom number generators with linear structures, K. Entacher, 1997". Retrieved 16 June 2012.
5. "How Visual Basic Generates Pseudo-Random Numbers for the RND Function".
6. Microsoft Support. Microsoft. Retrieved 17 June 2011.
7. In spite of documentation on MSDN, RtlUniform uses LCG, and not Lehmer's algorithm, implementations before Windows Vista are flawed, because the result of multiplication is cut to 32 bits, before modulo is applied
8. GNU Scientific Library: Other random number generators
9. Novice Forth library
10. Matsumoto, Makoto, and Takuji Nishimura (1998) ACM Transactions on Modeling and Computer Simulation
11. S.K. Park and K.W. Miller (1988). "Random Number Generators: Good Ones Are Hard To Find". Communications of the ACM31 (10): 1192–1201. doi:10.1145/63039.63042.
12. D. E. Knuth. The Art of Computer Programming, Volume 2: Seminumerical Algorithms, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89684-2. Section 3.2.1: The Linear Congruential Method, pp. 10–26.
13. P. L'Ecuyer (1999). "Tables of Linear Congruential Generators of Different Sizes and Good Lattice Structure". Mathematics of Computation68 (225): 249–260. doi:10.1090/S0025-5718-99-00996-5.

14. Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007), "Section 7.1.1. Some History", Numerical Recipes: The Art of Scientific Computing (3rd ed.), New York: Cambridge University Press, ISBN 978-0-521-88068-8
15. Gentle, James E., (2003). Random Number Generation and Monte Carlo Methods, 2nd edition, Springer, ISBN 0-387-00178-6.
16. Joan Boyar (1989). "Inferring sequences produced by pseudo-random number generators". Journal of the ACM36 (1): 129–141. doi:10.1145/58562.59305. (in this paper, efficient algorithms are given for inferring sequences produced by certain pseudo-random number generators).