

Congruencia Lineal

Implementar la congruencia lineal

```
In [4]: from tabulate import tabulate
from prettytable import PrettyTable
import pandas as pd
import numpy as np
import random
import math
xn=[]
un=[]
def congruencia(semilla,iteraciones,a,c,m,x):
    table = PrettyTable()
    table.field_names= ["# iteraccion","Xn","x","Un"]
    for i in range(1, iteraciones):
        xn = (a*x + c) % m; #FORMULA
        rn = xn/m
        x = xn
        table.add_row([i,xn,x,rn])
    #print(table)
    print(tabulate(table,tablefmt="fancy_grid"))

v=congruencia(15678,12,4343243,11,43,34321)
```

| # iteraccion | Xn | x | Un |
|--------------|----|----|----------------------|
| 1 | 35 | 35 | 0.813953488372093 |
| # iteraccion | Xn | x | Un |
| 2 | 2 | 2 | 0.046511627906976744 |
| # iteraccion | Xn | x | Un |
| 3 | 24 | 24 | 0.5581395348837209 |
| # iteraccion | Xn | x | Un |
| 4 | 38 | 38 | 0.8837209302325582 |
| # iteraccion | Xn | x | Un |
| 5 | 0 | 0 | 0.0 |
| # iteraccion | Xn | x | Un |
| 6 | 11 | 11 | 0.2558139534883721 |
| # iteraccion | Xn | x | Un |

| | | | | | | | | |
|---------|--------------|---------|----|---------|----|---------|---------------------|---------|
| | 7 | | 18 | | 18 | | 0.4186046511627907 | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | # iteraccion | | Xn | | x | | Un | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | 8 | | 42 | | 42 | | 0.9767441860465116 | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | # iteraccion | | Xn | | x | | Un | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | 9 | | 26 | | 26 | | 0.6046511627906976 | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | # iteraccion | | Xn | | x | | Un | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | 10 | | 8 | | 8 | | 0.18604651162790697 | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | # iteraccion | | Xn | | x | | Un | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |
| | 11 | | 20 | | 20 | | 0.46511627906976744 | |
| +-----+ | | +-----+ | | +-----+ | | +-----+ | | +-----+ |

```
In [9]: def congruencia(semilla,iteraciones,a,c,m,x):
        table = PrettyTable()
        table.field_names= ["# iteraccion", "Xn", "x0", "Un"]
        for i in range(1, iteraciones):
            xn = (a*x + c) % m; #FORMULA
            rn = xn/m
            x = xn
            table.add_row([i,xn,x,rn])
        #print(table)
        print(tabulate(table,tablefmt="fancy_grid"))
```

```
v=congruencia(3432234,8,5,3,9,1)
```

| # iteraccion | Xn | x0 | Un |
|--------------|----|----|--------------------|
| 1 | 8 | 8 | 0.8888888888888888 |
| 2 | 7 | 7 | 0.7777777777777778 |
| 3 | 2 | 2 | 0.2222222222222222 |
| 4 | 4 | 4 | 0.4444444444444444 |
| 5 | 5 | 5 | 0.5555555555555556 |
| 6 | 1 | 1 | 0.1111111111111111 |
| 7 | 8 | 8 | 0.8888888888888888 |

```

In [3]: iteraciones = int(input("Ingrese iteraciones: "))
print("Iter :", iteraciones)
seed = int(input("Ingrese semilla: "))
print("Xo:", seed)
a=int(input("Ingrese valor de a: "))
print("a:",a)
c=int(input("Ingrese valor de c: "))
print("c:",c)
m=int(input("Ingrese valor de m: "))
print("m:", m)
xn=[]
un=[]
def formula_conLineal(xo, A, C, M):
    form=((xo*A)+C)%M
    xn.append(form)
    return form

def dividido(n):
    d=n/m
    un.append(d)
    return d

xn.append(seed)
un.append(' ')
for i in range(iteraciones):
    valor=seed
    semilla=formula_conLineal(valor, a, c, m)
    dividido(seed)
    #table.add_row([xn,un])
#print(tabulate(table,tablefmt="fancy_grid"))
df=pd.DataFrame({"Xn":xn, "Un":un})
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
print(df)

```

```

Ingrese iteraciones: 45
Iter : 45
Ingrese semilla: 23
Xo: 23
Ingrese valor de a: 53
a: 53
Ingrese valor de c: 235
c: 235
Ingrese valor de m: 65
m: 65

```

| | Xn | Un |
|----|----|----------|
| 0 | 23 | |
| 1 | 24 | 0.353846 |
| 2 | 24 | 0.353846 |
| 3 | 24 | 0.353846 |
| 4 | 24 | 0.353846 |
| 5 | 24 | 0.353846 |
| 6 | 24 | 0.353846 |
| 7 | 24 | 0.353846 |
| 8 | 24 | 0.353846 |
| 9 | 24 | 0.353846 |
| 10 | 24 | 0.353846 |
| 11 | 24 | 0.353846 |
| 12 | 24 | 0.353846 |
| 13 | 24 | 0.353846 |
| 14 | 24 | 0.353846 |

| | | |
|----|----|----------|
| 15 | 24 | 0.353846 |
| 16 | 24 | 0.353846 |
| 17 | 24 | 0.353846 |
| 18 | 24 | 0.353846 |
| 19 | 24 | 0.353846 |
| 20 | 24 | 0.353846 |
| 21 | 24 | 0.353846 |
| 22 | 24 | 0.353846 |
| 23 | 24 | 0.353846 |
| 24 | 24 | 0.353846 |
| 25 | 24 | 0.353846 |
| 26 | 24 | 0.353846 |
| 27 | 24 | 0.353846 |
| 28 | 24 | 0.353846 |
| 29 | 24 | 0.353846 |
| 30 | 24 | 0.353846 |
| 31 | 24 | 0.353846 |
| 32 | 24 | 0.353846 |
| 33 | 24 | 0.353846 |
| 34 | 24 | 0.353846 |
| 35 | 24 | 0.353846 |
| 36 | 24 | 0.353846 |
| 37 | 24 | 0.353846 |
| 38 | 24 | 0.353846 |
| 39 | 24 | 0.353846 |
| 40 | 24 | 0.353846 |
| 41 | 24 | 0.353846 |
| 42 | 24 | 0.353846 |
| 43 | 24 | 0.353846 |
| 44 | 24 | 0.353846 |
| 45 | 24 | 0.353846 |

CONCLUSIONES

Un generador lineal congruencial (GLC) es un algoritmo que permite obtener una secuencia de números pseudoaleatorios calculados con una función lineal definida a trozos discontinua.