

## Procedural Content Generation Wiki


[Create account](#) or [Sign in](#)

Search this site

Search

[Articles](#) [Games](#) [Events](#) [Software](#) [Code](#) [Papers](#) [Images](#) [Community](#)  
[Links](#) [Pages](#) [Help](#) [Contact us](#)

# Linear Congruential Generator

[Most Reliable Cars](#)[Estate Planning Tips](#)[Chitika](#) | [Opt out?](#) 

## Description

The most common (and easiest to implement) [Pseudo Random Number Generator](#) is probably the Linear Congruential Generator.

The basic idea is to multiply the last number with a factor **a**, add a constant **c** and then modulate it by **m**.

Or as formula:  $X_{n+1} = (aX_n + c) \bmod m$ .

Where  $X_0$  is the seed.

## Code Example

Python example:

```
a = 3
c = 9
m = 16
xi = 0

def seed(x):
    global xi
    xi = x

def rng():
    global xi
    xi = (a*xi + c)%m
    return xi

for i in range(10):
    print rng()
```

Output:

```
9
4
5
8
1
12
13
0
9
4
```

Note that with the constants used in this example the generator has a period of 8.

[Linear congruential generator](#) - Wikipedia article.  
[z-rand.c](#) - Unangband source code, uses an LCG.

Previous: [Linear Congruential Generator](#)                      Next: [L-System](#)

[sequence generation](#)

page revision: 6, last edited: 20 Jul 2009, 09:36 (1757 days ago)

[Edit](#)   [Rate \(0\)](#)   [Tags](#)   [Discuss](#)   [History](#)   [Files](#)   [Print](#)   [Site tools](#)   [+ Options](#)

Powered by [Wikidot.com](#)                      [Help](#) | [Terms of Service](#) | [Privacy](#) | [Report a bug](#) | [Flag as objectionable](#)

Unless otherwise stated, the content of this page is licensed under [Creative Commons Attribution-ShareAlike 3.0 License](#)

Other interesting sites



**Neo Steam Wiki**  
May 2008 to February 2012



**Falchion Valley**  
Alpha Command Intra-Net



**CUNEF Group A2**  
For Your Work, Opinions and Discoveries!



**Igor**  
Independent Generation of Research