```
'''
        ================================================
            FEDERAL UNIVERSITY OF SANTA CATARINA
        ================================================

    File: ~/codigo/lcg.py
    Created on 4 de abr de 2017
    @author:  Rodrigo Pedro Marques
    GitHub: https://github.com/rodrigo93/INE5429-Trabalho1
    Professor: Renato Felipe Custodio

    This file is part of a college project for the INE5429 Computer Security
    course lectured in Federal University of Santa Catarina.

'''
import time

class LCG(object):
    """
        Linear Congruential Generator e um algoritmo gerador de numeros pseudo-aleatorios.
    """

    """
        Construtor da classe.
    """
    def __init__(self, semente):
        self.semente = semente
        return

    """
        Formula do algoritmo LCG para gerar os numeros pseudo-aleatorios.
    """
    def gerador(self, m, a, c):
        self.semente = (a*self.semente + c) % (2**m)
        return self.semente

    """
        Metodo utilizado para testar o algoritmo LCG.
        Como exemplo, utilizei os valores:
        semente = 74573,
        m = aos tamanhos especificados no enunciado
        a = 1103515245
        c = 12345
    """
    def teste(self):
        outFile = open("lgc_output.txt", "wb")

        tamanhos = [40, 56, 80, 128, 168, 224, 256, 512, 1024, 2048, 4096]
        tabelaDeResultado = []
        indice = 0;
        for m in tamanhos:
```

```
52              indice += 1
53              tabelaDeResultado.append(self.gerador(m, 1103515245, 12345))
54              print "Para o tamanho m = ", m, " gerou-se o numero ",
  tabelaDeResultado[indice-1]
55              outFile.write(str(tabelaDeResultado[indice-1]) + "\n")
56
57          outFile.close()
58          return
59
60      """
61          Eh importante lembrar as regras:
62
63          m > 0, modulo,
64          0 < a < m, multiplicador,
65          0 <= c < m, incrementador,
66          0 <= semente < m, valor inicial
67
68          Para o LCG ser um "mixed generator" deve-se respeitar as seguintes
  regras:
69          'm' e 'c' sao relativamente primos;
70          'a-1' eh divisivel por todos os fatores primos de 'm';
71          'a-1' eh divisivel por 4 se o 'm' tambem for
72      """
73
74  """
75      Funcao inicial
76  """
77  if __name__ == '__main__':
78      start_time = time.time()
79      lcg = LCG(74573)
80      lcg.teste()
81      print("--- Tempo de execucao: %s segundos ---" % (time.time() -
  start_time))
82
83
84
```