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

    File: ~/codigo/blumblumshub.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 BBS(object):
    """
        Blum Blum Shub e um algoritmo gerador de numeros pseudo-aleatorios.
    """

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

    """
        Formula do algoritmo BBS para gerar os numeros pseudo-aleatorios.
        Ele apenas recebe o modulo 'm' e gera um numero
    """
    def gerador(self, m):
        self.seed = (self.seed**2) % (2**m)
        return self.seed

    """
        Metodo utilizado para testar o algoritmo Blum Blum Shub.
    """
    def teste(self):
        outFile = open("bbs_output.txt", "wb")

        tamanhos = [40, 56, 80, 128, 168, 224, 256, 512, 1024, 2048, 4096]
        tabelaDeResultado = []
        indice = 0;
        for m in tamanhos:
            indice += 1
            tabelaDeResultado.append(self.gerador(m))
            print "Para o tamanho m = ", m, " gerou-se o numero ", tabelaDeResultado[indice-1]
            outFile.write(str(tabelaDeResultado[indice-1]) + "\n")
```

```
52
53          outFile.close()
54          return
55
56  """
57      Funcao inicial
58  """
59  if __name__ == '__main__':
60      start_time = time.time()
61      bbs = BBS(88667)
62      bbs.teste()
63      print("--- Tempo de execucao: %s segundos ---" % (time.time() -
    start_time))
```