Trabalho RSA

(1) Implementar um programa para criptografia e descriptografia de strings informadas pelo usuário usando o algoritmo RSA usando um número variável de bits para a chave criptográfica (usar potências de dois: 8, 16, 32, 64, 128, ...);

(2) Implementar um algoritmo de força bruta para quebrar a chave criptográfica – fatoração da chave pública nos números primos que a geraram – e verificar o tempo levado para quebrar as chaves de diferentes tamanhos;

(3) Elaborar um relatório técnico de até sete páginas no formato de artigo da SBC, descrevendo o funcionamento de criptografia de chave pública RSA e o método usado na sua implementação (sem apresentar código fonte), apresentando a complexidade das principais funções implementadas, e um gráfico com os tempos de execução da geração das chaves e do tempo levado para fatorar a chave em inteiros com *n*-bits (i.e o tempo para decifrar a criptografia).

Linguagens permitidas: C, C++, Java, Python ou Haskell.

Pode-se utilizar as bibliotecas (importadas ou nativas) para manipular números grandes (BigInt), contudo, é **obrigatória** a implementação das seguintes funções:

- Geração das chaves pública e privadas:
  - Função para gerar primos pequenos (crivo de eratóstenes)
  - Função para verificação do caráter pseudoprimo usando Miller-Rabin simplificado (sem verificar por raízes quadradas não triviais de 1 módulo n);
  - Algoritmo de inverso modular
  - Algoritmo de euclides extendido;
  - Algoritmo de exponenciação modular;

- Função para criptografar e descriptografar uma String dada como entrada pelo usuário. Deve-se criptografar a mensagem em blocos de *n*-caracteres;

- Algoritmo de força bruta para a fatoração da chave pública nos números primos que a geraram.