QUESTÃO 01 (1,0 ponto)

Projeto de Programação: Sistema RSA com Fatoração ρ de Pollard e Aplicação de Teoremas Modulares em Três Etapas.

Objetivo:
Implementar em C ou C++ um sistema completo de criptografia e descriptografia RSA, iniciando pela fatoração de números compostos usando o método ρ de Pollard, e aplicando corretamente conceitos de aritmética modular (como o Teorema de Fermat, o Teorema de Euler e a Divisão Euclidiana) para os cálculos de potência modular durante a codificação e decodificação de mensagens.

Etapa 1: Fatoração Interativa (Método ρ de Pollard)

Objetivo: Descobrir os fatores primos p e q de dois números compostos N_1 e N_2.

Entrada de dados:

O programa deve solicitar **dois números compostos distintos** N_1 e N_2.

Restrição: Cada número deve possuir **3 ou 4** dígitos, ou seja, entre 100 e 9999.

Informe ao usuário que cada N_i deve ser produto de **primos distintos** para que o método ρ de Pollard seja eficiente.

Implementação do método ρ de Pollard:

Utilize a função de iteração: $g(x)=(x^2+1) \bmod N_i$

Semente – $x_0=2$.

Em cada iteração, calcule: mdc ($|x_2-x_1|$,N_i) até encontrar um fator p_i não trivial de N_i

O programa deve exibir **cada passo da iteração**.

Definição dos primos RSA:

Seja p o fator encontrado de N_1

Seja q o fator encontrado de N_2.

Exiba claramente os valores de p e q.

****Observação:** O cálculo do mdc deve ser feito **utilizando o Algoritmo de Euclides**, implementado pelo aluno (não é permitido usar funções prontas como std::gcd).

Etapa 2: -Geração das Chaves RSA

Objetivo: Construir o par de chaves pública e privada do sistema RSA.

Cálculo do módulo: $n = p \times q$

Totiente de Euler: $z(n) = (p-1) \times (q-1)$

Escolha do expoente público: Escolha o menor $E > 1$ e $E < n$ tal que mdc $(E , z(n))=1$

Cálculo do expoente privado: Encontre D tal que: $D \times E \equiv 1 \bmod z$

*Utilize o **Algoritmo Estendido de Euclides** para determinar o inverso modular de E em relação a z

Impressão das chaves:

Chave pública: (n,e)

Chave privada: (n,d)

Etapa 3 - Codificação (Criptografia) e Decodificação (Descriptografia)

Objetivo: Realizar a criptografia e a decodificação de uma mensagem, aplicando o teorema modular adequado e um sistema próprio de codificação numérica de letras.

Pré – Codificação

Antes de aplicar a criptografia RSA, cada caractere da mensagem deve ser convertido em um número segundo o sistema de pré-codificação do alfabeto: A = 11, B= 12, …, Z= 36. Espaço = 00.

Codificação

Para cada bloco M formado pelos números da mensagem: $C \equiv M^E \pmod n$

O programa deve exibir **o cálculo passo a passo** da exponenciação modular.

Decodificação

Para cada bloco cifrado C: $M \equiv C^D \pmod n$

O resultado M deve ser reconvertido para letras segundo a tabela de pré-codificação.

* Lembre-se cada bloco será referente a apenas 2 dígitos.

Resolução da exponenciação modular

Durante o cálculo de $M^E \pmod n$ e $C^D \pmod n$, o programa deve:

- Verificar as condições e selecionar automaticamente o método de redução de expoente:
    - **Pequeno Teorema de Fermat**, se n for primo;
    - **Teorema de Euler**, se mdc(M,n)=1;
    - **Teorema da Divisão Euclidiana**, para reduzir o expoente.
- O programa deve indicar na saída textual qual teorema foi aplicado e mostrar o cálculo correspondente.

Observações

- Espaços e pontuações podem ser ignorados ou substituídos por um código fixo (exemplo: 00 para espaço).
- O código deve ser implementado **em C ou C++**, sem uso de bibliotecas externas de criptografia.
- Todas as funções fundamentais (como cálculo de mdc, inverso modular, e exponenciação modular) devem ser **programadas pelo aluno**.
- O programa deve **imprimir o passo a passo de todos os pontos principais do cálculo**, incluindo:
    1. Iterações do método ρ de Pollard;
    2. Cálculo do mdc (Algoritmo de Euclides);
    3. Determinação do inverso modular (Euclides Estendido);
    4. Escolha e aplicação do teorema modular (Fermat, Euler ou Divisão Euclidiana);
    5. Processo completo de criptografia e descriptografia;
    6. Reconversão numérica em texto.
- O sistema deve confirmar que a mensagem decifrada é idêntica à mensagem original.
- Os alunos devem comentar no código as decisões tomadas e justificar o método modular escolhido em cada etapa.

QUESTÃO 02 (0,5 ponto)