

## Introdução à Computação Prof. Bernardo

### 6º Exercício: Criptografando arquivo com algoritmo RSA

Elabore um programa para realização de Criptografia de um arquivo texto (“mensagem.txt”) base no algoritmo RSA. No texto a ser criptografado serão utilizadas apenas letras do alfabeto observando tabela abaixo com os respectivos valores:

1	2	3	4	5	6	7	8	9	10	11
A	B	C	D	E	F	G	H	I	J	L

12	13	14	15	16	17	18	19	20	21	22	23
M	N	O	P	Q	R	S	T	U	V	X	Z

Tabela 01 Letras e valores correspondentes

O programa irá solicitar que o usuário digite dois números primos menores que 100 para gerações do código criptografado. Entradas erradas deverão ser informadas e não serão aceitas pelo programa.

A partir dos valores digitados para **p** e **q** obter **n**,  $n=p*q$ . **n** será o tamanho do conjunto.

$$Z_n = \{0,1,2,3,4,5,6,\dots,n\}$$

O próximo passo será o cálculo da função Totiente de Euler,

$$\varphi(x) = (p - 1) * (q - 1)$$

Após vamos escolher o menor **e** > 1, cujo  $\text{MDC}(\varphi(x), e) = 1$

Este valor será a chave pública que será usado na criptografia do texto.

$$c = m^e \bmod n$$

onde **e** é a chave pública e **m** é o valor numérico da letra. **C** será o valor da letra criptografada.

Por exemplo cifrando a letra T teremos:

$$T = 19$$

Para  $p = 7$  e  $q = 5$  escolhidos pelo usuário:

$$n = 35$$

$$\varphi(x) = 6 \cdot 4 = 24$$

$$\text{MDC}(24, 2) = 2;$$

$$\text{MDC}(24, 3) = 3;$$

$$\text{MDC}(24, 4) = 4;$$

$$\text{MDC}(24, 5) = 1;$$

$$e = 5$$

$$c = 19^5 \bmod 35$$

$$c = 24$$

O texto a ser criptografado será escrito em “caixa baixa” (minúsculas) ou caixa alta apenas com letras, espaços devem ser ignorados, porém o programa deverá fazer a conversão para caixa alta para efeito de conversão visando utilização da tabela 01. Após a criptografia do texto, o programa deverá criar um arquivo de saída com os valores de cada letra separados por um espaço. Por exemplo a palavra TAL ficaria:

mensagem.txt -> TAL

mensagemCriptografada.txt -> 24 1 16

Para decifrar o texto, a fórmula a ser usada será:

$$m = c^d \bmod n$$

Esta fórmula é obtida pela repetição sucessiva do algoritmo de Euclides, porém para o laboratório, ao invés de calcular a chave privada, vamos tentar achá-la via força bruta. Com base no valor das letras  $p$  e  $q$  já conhecidos, realizar um laço de iteração até encontrar o valor **d** que satisfaz o algoritmo. Começar com a valor 2 até o limite do valor máximo do tipo **unsigned int**. Apresente o

resultado em forma de tabela com o número de tentativas necessárias para descoberta da chave privada. Faça também o cálculo para os próximos dois valores sequenciais possíveis para a chave pública e colocando na mesma tabela anterior.

Como saída final do programa imprimir na tela o arquivo decifrado no vídeo com todas as letras na sequência que foram lidas no arquivo mensagem.txt porém sem espaços e em caixa baixa.

#### Observações:

- 1-Trabalhe com funções para organizar o código
- 2-Crie uma lista ligada para armazenar os dados lidos no arquivo de entrada. Cada nó deverá armazenar um tipo char e um tipo int (valor resultante da criptografia).
- 3- Utilize as boas práticas de programação ensinadas em sala de aula.
- 4- Referência para os cálculos no endereço:  
<https://www.lambda3.com.br/2012/12/entendendo-de-verdade-a-criptografia-rsa-parte-ii/>

#### Entregar:

Arquivos SeuNome\_lab06.cpp e SeuNome\_lab06.exe utilizando o sistema TIDIA no menu Atividades -> Lab06.

#### Cabeçalho:

Obrigatoriamente, **no início** do arquivo fonte, coloque um cabeçalho na forma:

```
/* Copyright by SEU NOME */
/* Turma 3 */
/* Lab 06: Criptografia em arquivo texto */
/* Programa compilado com CodeBlocks 17.02 */
```