

Potência de Números Grandes

Mário Leite

...

A Criptografia é o método mais eficiente para a proteção de dados/informações. Entre os métodos mais utilizados para codificar uma mensagem está o chamado **“Método RSA”**, que implementa uma solução baseada em chave dupla (assimétricas), fornecendo números que *codificam e decodificam* a informação. Neste caso, o emissor envia a mensagem criptografada (*codificada*) com uma chave: a *chave pública*; e para ler esta mensagem o receptor deve ter uma outra chave: a *chave privada*. É como se existissem duas chaves físicas para a fechadura de uma mesma porta: uma para *trancar* e outra para *destrancar*. Estas duas chaves são criadas com base no produto de dois números primos muito grandes! É aí que reside a dificuldade de “quebrar” a codificação (decodificar a mensagem) baseada no método RSA. Este método recomenda que os dois números primos escolhidos tenham cerca de, no mínimo, 100 dígitos; ou como se diz: deve-se trabalhar com chaves a partir de 330 *bits*. A decodificação envolve, basicamente, a fatoração desses números primos para se obter a função de Euler. E neste caso o processamento para fatorar um número primo muito grande pode durar até *milhões* de anos em sistemas computacionais convencionais; não quânticos. Por isto, diz-se que é quase impossível decodificar um texto codificado com o algoritmo RSA. Entretanto, um problema inerente a este algoritmo é com relação ao cálculo modular do tipo: **($x \bmod y$)**. Por exemplo, sendo $x=8031810176$ (x^7) e $y=1062637$, o resultado é 399730. Observe que é um cálculo muito “pesado”, mesmo para valores pequenos de **x** e **y**, comparados com os valores recomendados pelo método. Mas, tal aparente dificuldade pode ser facilmente resolvida com o programa **“PotenciaModular”** codificado aqui em Visualg, e que gastou menos de 20 milissegundos para apresentar o resultado!

Para adquirir o *pdf/e-book* deste livro ou o *pdf* de outros livros sobre programação, entre em contato pelo *e-mail*: **marleite@gmail.com**

Algoritmo "PotenciadModular"

//Faz cálculo modular com potência de números grandes
//Em Visualg
//Autor: Mário Leite

Var R, j: inteiro
a, x, n: inteiro

Inicio

x <- 7
n <- 1062637
a <- 26
R <- 1

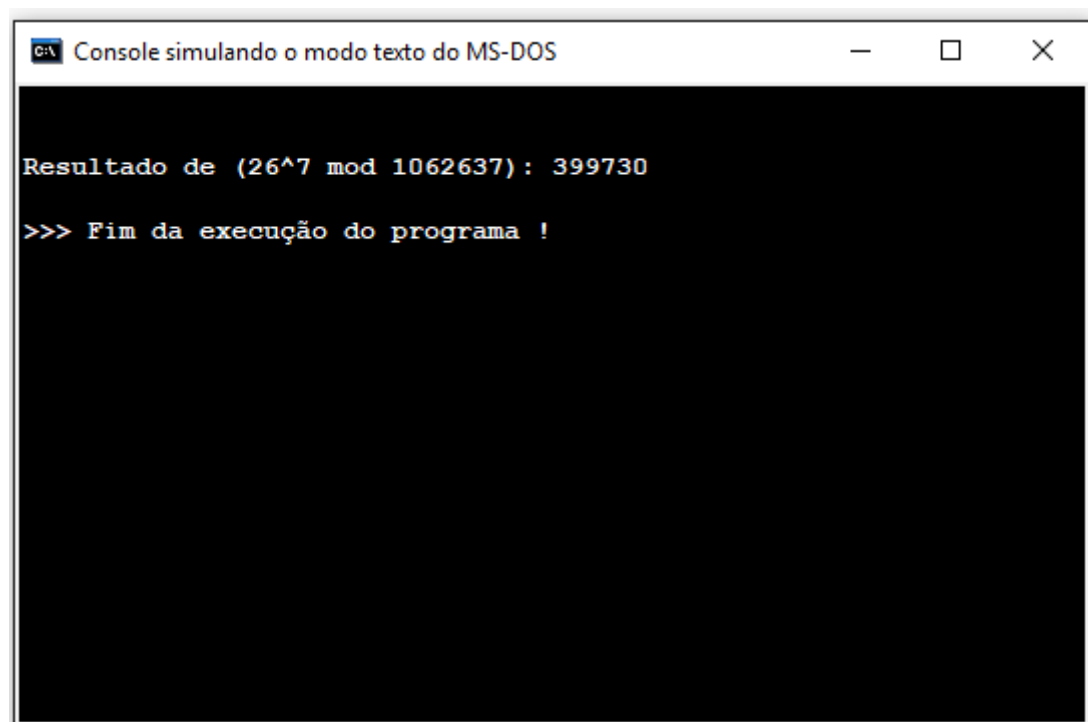
Para j **de** 1 **Ate** x **Faca**
R <- (R*a) **Mod** n
FimPara

Este bloco de instruções é que faz o trabalho "pesado" de cálculo modular.

Escreval ("")

Escreval ("Resultado de (26^7 mod 1062637): ", R)

FimAlgoritmo



The screenshot shows a window titled "Console simulando o modo texto do MS-DOS". The window has a black background with white text. The text displayed is: "Resultado de (26^7 mod 1062637): 399730" followed by a blank line and then ">>> Fim da execução do programa !".

```
C:\> Console simulando o modo texto do MS-DOS

Resultado de (26^7 mod 1062637): 399730

>>> Fim da execução do programa !
```