Programação I e Lógica e Técnica de Programação

T5 - Trabalho 5

Prof.: Paulo Roberto Nunes de Souza

1 Introdução

Este trabalho cobre o assunto de vetores, strings e for. O trabalho é para ser feito em **grupo de até 3 pessoas** e deve ser entregue pelo Google Sala de Aula da disciplina.

2 Mensagem codificada

Codificar uma mensagem consiste em embaralhar sua informação de tal forma que, pessoas que não tenham o conhecimento de como desembaralhar as informações não entendam a mensagem, porém as pessoas que tenham esse conhecimento consigam desembaralhar e entender a mensagem com pouco esforço.

Formas mais recentes de codificar mensagem centralizam o conhecimento crucial para embaralhar e desembaralhar a mensagem numa sequencia de caracteres ou números conhecido como **chave**. Com isso o método de embaralhamento pode ser conhecido de todos, mas só quem conhece a **chave** a ser usada consegue corretamente embaralhar e desembaralhar a mensagem.

3 Método de codificação

O método a ser usado utiliza uma operação matemática simples aplicada entre cada letra da mensagem e da chave progressivamente. A operação matemática que será utilizada na codificação chama-se *XOR* e matematicamente é simbolizada por $\oplus$ e na linguagem C é executado utilizando o símbolo de acento circunflexo ($\hat{}$).

Consideremos que a mensagem original será simbolizada por M, a chave será simbolizada por K e a mensagem codificada será simbolizada por C, sendo que:

$$M = \{m_1, m_2, m_3, \ldots, m_n\}$$
$$C = \{c_1, c_2, c_3, \ldots, c_n\}$$
$$K = \{k_1, k_2, k_3, \ldots, k_q\}$$

3.1 Codificação

Quando estamos de posse da mensagem M e da chave K e temos o interesse de gerar a mensagem codificada C, este processo é chamado de codificação. Este processo é feito aplicando a operação *XOR*, uma a uma, entre cada letra da mensagem M e cada letra da chave K, gerando assim cada letra da mensagem codificada C. Este processo pode ser representado matematicamente da seguinte forma.

$$\begin{array}{cccccccccc} m_1 & m_2 & \ldots & m_{i-1} & m_i & m_{i+1} & m_{i+2} & \ldots & m_{n-1} & m_n \\ \oplus & \oplus & \ldots & \oplus & \oplus & \oplus & \oplus & \ldots & \oplus & \oplus \\ k_1 & k_2 & \ldots & k_{q-1} & k_q & k_1 & k_2 & \ldots & k_{j-1} & k_j \\ \hline c_1 & c_2 & \ldots & c_{i-1} & c_i & c_{i+1} & c_{i+2} & \ldots & c_{n-1} & c_n \end{array}$$

Onde:

n**:** Tamanho da mensagem M e da mensagem codificada C;

q**:** Tamanho da chave K;

i**:** Valor arbitrário entre 1 e n. $1 < i < n$;

j**:** Valor arbitrário entre 1 e q. $1 < j < q$.

Ao final o processo terá calculado cada caracter da mensagem codificada C.

$$c_i = m_i \oplus k_j$$

Observem que, ao codificar cada caracter da mensagem, pode acontecer da chave chegar ao fim. Quando isso acontecer, o processo continua utilizando o primeiro caracter da chave novamente e partindo daí em diante sempre que necessário.

3.2 Decodificação

Quando estamos de posse da mensagem codificada C e da chave K e temos o interesse de gerar a mensagem M, este processo é chamado de decodificação. Este processo é feito aplicando a mesma operação *XOR*, uma a uma, entre cada letra da mensagem codificada C e cada letra da chave K, gerando assim cada letra da mensagem M. Este processo pode ser representado matematicamente da mesma forma que a codificação, só que aplicando-se à mensagem codificada C para se obter a mensagem M.

c_1	c_2	...	c_{i-1}	c_i	c_{i+1}	c_{i+2}	...	c_{n-1}	c_n
$\oplus$	$\oplus$	...	$\oplus$	$\oplus$	$\oplus$	$\oplus$	...	$\oplus$	$\oplus$
k_1	k_2	...	k_{q-1}	k_q	k_1	k_2	...	k_{j-1}	k_j
m_1	m_2	...	m_{i-1}	m_i	m_{i+1}	m_{i+2}	...	m_{n-1}	m_n

Onde:

n**:** Tamanho da mensagem M e da mensagem codificada C;

q**:** Tamanho da chave K;

i**:** Valor arbitrário entre 1 e n. $1 < i < n$;

j**:** Valor arbitrário entre 1 e q. $1 < j < q$.

Ao final o processo terá calculado cada caracter da mensagem M.

$$m_i = c_i \oplus k_j$$

3.3 Exemplo de codificação

Supondo que o valor da mensagem M e da chave K sejam $\boldsymbol{M}$ **= "Atividade EARTE"** e $\boldsymbol{K}$ **= "Segredo"**, a codificação seria a seguinte.

'A'	't'	'i'	'v'	'i'	'd'	'a'	'd'	'e'	' '	'E'	'A'	'R'	'T'	'E'
$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$	$\oplus$
'S'	'e'	'g'	'r'	'e'	'd'	'o'	'S'	'e'	'g'	'r'	'e'	'd'	'o'	'S'
18	17	14	4	12	0	14	55	0	71	55	36	54	59	22

Com isso produzimos a mensagem codificada C, que neste exemplo tem o seguinte valor. $\boldsymbol{M}$ **= {18, 17, 14, 4, 12, 0, 14, 55, 0, 71, 55, 36, 54, 59, 22}**.

18	17	14	4	12	0	14	55	0	71	55	36	54	59	22
⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'S'	'e'	'g'	'r'	'e'	'd'	'o'	'S'	'e'	'g'	'r'	'e'	'd'	'o'	'S'
'A'	't'	'i'	'v'	'i'	'd'	'a'	'd'	'e'	' '	'E'	'A'	'R'	'T'	'E'

3.4 Exemplo de decodificação

Supondo que o valor da mensagem codificada C e da chave K sejam $\boldsymbol{M}$ **= {18, 17, 14, 4, 12, 0, 14, 55, 0, 71, 55, 36, 54, 59, 22}** e $\boldsymbol{K}$ **= "Segredo"**, a decodificação seria a seguinte.

Com isso produzimos a mensagem M, que neste exemplo tem o seguinte valor. $\boldsymbol{M}$ **= "Atividade EARTE"**.

4 Programa

O programa deverá implementar tanto a codificação quanto a decodificação.

Para permitir que o usuário acesse a opção que ele deseja, o programa deve primeiramente mostrar na tela um menu com 3 opções:

1 Codificar;

2 Decodificar;

3 Sair.

Após apresentado o menu, o programa deve pedir para o usuário digitar a opção escolhida (1, 2, ou 3). Caso o usuário tenha escolhido a opção 1 ou 2, o programa deverá então executar a função escolhida pelo usuário e mostrar o menu novamente ao final. Caso o usuário tenha escolhido a opção 3, o programa deverá terminar.

4.1 Opções codificação e decodificação

Nas operações de codificação e decodificação a mensagem M deve aceitar mensagem de até 60 caracteres, não necessitando ser capaz de manipular mensagens maiores. Da mesma forma, a mensagem codificada C deve aceitar mensagem de até 60 números inteiros, não necessitando ser capaz de manipular mensagens maiores.

Ao iniciar a codificação, o programa deverá solicitar que o usuário digite a mensagem M sem espaços, para codificá-la. Após receber a mensagem e fazer os cálculos correspondentes, o programa deverá imprimir na tela o tamanho da mensagem digitada e a mensagem codificada C correspondente.

Ao iniciar a decodificação, o programa deverá solicitar que o usuário primeiramente digite o tamanho da mensagem que será digitada, em seguida o programa deve solicitar que o usuário digite a mensagem codificada C, para decodificá-la. Após receber a mensagem e fazer os cálculos correspondentes, o programa deverá imprimir na tela mensagem M correspondente.

A mensagem M deverá sempre ser tratada, tanto para leitura quanto para impressão, como uma sequência (vetor) de caracteres. Já a mensagem codificada C deverá sempre ser tratada, tanto para leitura quanto para impressão, como uma sequência (vetor) de números inteiros.

A chave a ser utilizada em ambos os processos será $\boldsymbol{K}$ **= "EARTE2020/1"**.

4.2 Lembretes

- A operação *XOR*, que matematicamente é simbolizada por $\oplus$, na linguagem C é executado utilizando o símbolo de acento circunflexo ($\hat{}$);
- Para obter o tamanho da mensagem digitada pelo usuário, pode ser utilizada a função *strlen* da biblioteca *string.h*;

- A chave a ser utilizada em ambos os processos será $\boldsymbol{K}$ **= "EARTE2020/1"**;
- A mensagem M deverá sempre ser tratada como uma sequência (vetor) de caracteres;
- A mensagem M não poderá ter espaços;
- Antes de digitar a mensagem codificada C, o usuário deverá digitar qual é o tamanho desta mensagem.
- A mensagem codificada C deverá sempre ser tratada como uma sequência (vetor) de números inteiros;
- O trabalho é para ser feito individualmente ou em dupla.

5 Formato de entrega

Os alunos devem entregar o código fonte em linguagem C que resolve o problema proposto num arquivo no formato .c. Atividades entregues em formato PDF, DOC, JPG, PNG, ZIP, URL e etc, terão pontuação descontada por não se adequarem ao solicitado no trabalho.

Informações extras que forem colocadas no arquivo como nome, número de matrícula, curso e etc, devem ser colocados como comentário. Caso não seja feito conforme solicitado o código certamente gerará um erro de compilação, o que acarretará em perda de pontuação na atividade.

6 Entrega

A entrega deve ser feita pelo Google Sala de Aula da disciplina. Entregas feitas após o prazo do trabalho serão penalizadas conforme especificado no Plano de Ensino da disciplina.