



Campus Colatina – Bacharelado em Sistemas de Informação

Disciplina: **Programação II**

Professor: **Giovany Frossard Teixeira**

Observações: 1º Trabalho (20 pontos)

Data:

Especificação

Contextualização:

A presidente Dilma Rousseff determinou que a inteligência brasileira elaborasse um sistema de criptografia de mensagens que não fosse quebrado pelos americanos. Você então foi incumbido dessa tarefa. O sistema, a ser elaborado, deve ter a possibilidade de criar mensagens e decodificá-las.

As mensagens enviadas são sempre em formato de string, ou seja, consistem num vetor de caracteres. Para simplificação serão usados apenas caracteres em minúsculo, dígitos e os símbolos ‘,’ (vírgula) e ‘.’ (ponto).

A lei de formação para a criptografia das mensagens é a seguinte:

Pré processamento:

Dobrando a meta: toda vez que for encontrada a substring “meta” ela será substituída por “dobrarmeta”. Essa técnica deve ser aplicada a cada uma das substrings da string geral, se a substring for diferente de “meta” ela é apenas copiada. (suponha que não existe a substring “dobrarmeta” na string original)

Ex: “nao vamos ter meta, vamos dobrar a meta” =>
“nao vamos ter dobrarmeta, vamos dobrar a dobrarmeta”

1. A criptografia é aplicada a cada substring da string definida. Sendo que uma substring é dada pelo conjunto de caracteres até que se encontre um espaço em branco ou o final da string. Ex: “o rato roeu a roupa do rei de roma” existem 9 substrings.
2. Existem 3 padrões de criptografia que serão aplicados nas substrings:

- a. **Enxerto de tamanho:** conta o número (num) de caracteres da substring e avança <num> caracteres nos caracteres. Caso o character passe do ASC de 'z' deve ser atribuído o ASC 'z' e alguma informação adicional pode ser armazenada para a descryptografia (um vetor de números inteiros pode ser gerado e usado como apoio, por exemplo, para a descryptografia).

Ex1: "azera" => "fzjxf"

num = 5

('a' + 5 = 'f', 'z' = 'z' sobram 5, 'e' + 5 = 'j', 'r' + 5 = 'x', 'a' + 5 = 'f')

Sugestão: Gere um vetor de inteiros com os valores {0, 5, 0, 0, 0}

Ex: "xa" => "zc"

num = 2

('x' + 2 = 'z' sobra 1, 'a' + 2 = 'c')

Sugestão: Gere um vetor de inteiros com os valores {1, 0}

- b. **Números maus:** se forem encontrados os dígitos 6 ou 9 na substring, eles devem ser "jogados" para o início (6) ou para o final (9) da substring.

Ex: "ab6cd9696xx6933" => "6666abcdxx33999"

Sugestão: Guarde as posições dos 6s e dos 9s da substring original para a descryptografia v1 = {2, 6, 8, 11} v2 = {5, 7, 12}

- c. **Max letra:** procura a maior letra ('z' é a maior de todas e 'a' a menor) e dobra essa letra onde ela estiver.

Ex1: "aa" => "aaaa" (se só tem 'a' ela é a maior de todas então dobramos ambos 'a's.).

Ex2: "abc" => "abcc" (dobramos apenas o 'c' pois ele é o maior)

Ex3: "bca" => "bccca" (dobramos apenas o 'c' pois ele é o maior)

Ex4: "ccaaccbb" => "ccccaacccbb" (dobramos todos os 'c's.).

3. A ordem de aplicação dos padrões de criptografia nas substrings da string principal é: Enxerto de tamanho, Números maus e Max letra **em seguida começa-se o processo novamente**. É importante notar que antes de aplicar os padrões é necessário fazer o pré-processamento (Dobrando a meta).

- a. Ex1: "nao va9mo6s meta ter" => "nao va9mo6s dobrarmeta ter" =>

"nao" => "qdr"

"va9mo6s" => "6vamos9"

"dobrarmeta" => dobrarmetta"

"ter" => "xhu"

⇒ “qdr 6vamos9 dobrarmetta xhu” (string criptografada)

- 1) (1 ponto) Faça a função **lerString** que aloca dinamicamente uma string de 5000 caracteres (use uma constante `#define TAMANHO 5000`) e promove a leitura da mesma pelo teclado, retornando a string informada.

A assinatura da função é: *char* lerDados()*

- 2) (2 pontos) Faça a função **dobrarMeta** que recebe a string informada e retorna uma nova string pré-processada.
- 3) (7 pontos) Faça a função **criptografaDados** que recebe a string informada e retorna uma nova string com os dados criptografados.
- 4) (8 pontos) Faça a função **descriptografaDados** que recebe uma string criptografada e recria a mensagem original retornando-a.
- 5) (2 pontos) Faça a função **voltarMeta** que recebe a string pré-processada (que passou por **dobrarMeta**) e retorna a string original (“dobrarmeta” => “meta”)

O main() deve ficar mais ou menos assim:

```
int main(){
    char *palavra = lerDados();
    char *palavra_pre_processada = dobrarMeta(palavra);
    char *palavra_criptografada = criptografaDados(palavra_pre_processada,
<possíveis parâmetros adicionais para apoio a descriptografia>);
    printf(“A string criptografada é: %s\n”, palavra_criptografada);
    char *palavra_original_pre_processada = descriptografia(palavra_criptografada,
<possíveis parâmetros opcionais>);
    char *palavra_original = voltarMeta(palavra_original_pre_processada);
    printf(“A string original é: %s\n”, palavra_original);
}
```

Exemplo de entrada de dados:

Digite a string de mensagem:

nao va9mo6s meta ter (**digitado pelo usuário**)

A string criptografada é: qdr 6vamos9 dobrarmetta xhu

A string original é: nao va9mo6s meta ter

Importante: Na impressão da string original não é permitido informar a string lida é necessário chamar a função de descriptografia.

Data de Entrega: 26/10/2015

Componentes por Grupo: 2 (máximo)

Será feita entrevista para validação da implementação.

Valor: 20 pontos (20 % da nota final). A nota da dupla é a nota do aluno que pontuar menos, essa abordagem visa forçar a dupla a realmente trabalhar em conjunto.

OBS1. NÃO SERÃO ACEITOS trabalhos entregues após essa data, sendo computado para tanto o VALOR 0 à nota.

Bom Trabalho !!!