



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:

Eliminando a corrupção: toda vez que for encontrada a substring “corrupção” ela será substituída por “”, ou seja, a “corrupção” é eliminada da string original. Essa técnica deve ser aplicada a cada uma das substrings da string geral, se a substring for diferente de “corrupção” ela é apenas copiada.

Ex: “nunca antes na história desse país tanto se combateu a corrupção, mas também nunca foram encontrados tantos casos de corrupção”

=>

“nunca antes na história desse país tanto se combateu a , mas também nunca foram encontrados tantos casos de ”

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. **Dança das letras:** primeiramente conta o número (num) de caracteres da substring. Depois se calcula a raiz quadrada desse valor (raiz quadrada de num). Essa raiz quadrada deverá ser o número de trocas dos pares “quase” consecutivos das letras da string. Letras “quase” consecutivas tem uma letra entre elas.

Ex1: “a” => 1 letra => raiz quadrada = 1 => 1 troca, mas só há um caracter então não há o que trocar → “a”

Ex2: “ab” => 2 letras => raiz quadrada = 1.4 => 1 troca, mas só existem 2 caracteres então não há o que trocar → “ab”

Ex3: “abc” => 3 letras => raiz quadrada = 1.73 => 1 troca, troca-se ‘a’ por ‘c’ → “cba”

Ex4: “abcd” => 4 letras => raiz quadrada = 2 => 2 trocas, troca-se ‘a’ por ‘c’ e ‘b’ por ‘d’ → “cdab”

Ex5: “abcde” => 5 letras => raiz quadrada = 2.2 => 2 trocas, troca-se ‘a’ por ‘c’ e ‘b’ por ‘d’ → “cdabe”

Ex6: “abcdefghijklmnop” => 15 letras => raiz quadrada = 3.87 => 3 trocas, troca-se ‘a’ por ‘c’, ‘b’ por ‘d’ e ‘e’ por ‘a’ → “cdebafghijklmnop”

- b. **Sono ampliado:** toda vez que forem encontrados os caracteres “zz” eles devem ser dobrados no local encontrado e colocados no inicio e no final da substring.

Ex1: “abacazzztado” → “zzzzabacazzzzzzzzztadozzzz”

Ex2: “anazta” → “anazta”

Ex3: “anazzta” → “zzanazzzztazz”

- c. **Letra b “Maligna”:** toda vez que a letra ‘b’ for encontrada os 2 caracteres seguintes devem ser eliminados da substring (mesmo se um deles for um ‘b’).

Ex1: “batata” => “bata”

Ex2: “babaca” => “baca”

Ex3: “baabxxbccd” => “bbbd”

3. A ordem de aplicação dos padrões de criptografia nas substrings da string principal é: *Dança das letras, Sono ampliado e Letra b “Maligna”* **em seguida começa-se o processo novamente**. É importante notar que antes de aplicar os padrões é necessário fazer o pré-processamento (*Eliminando a corrupção*).

- a. Ex1: “abcdefghijklmnop corrupção abacazzztado baabxxbccd corrupção abcd”
⇒ “abcdefghijklmnop abacazzztado baabxxbccd abcd”
⇒ “cdebafghijlmnop zzzzabacazzzzzzzzztadozzzz bbbd cdab” (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 **eliminarCorrupcao** 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 **recriarCorrupcao** que recebe a string pré-processada e recoloca a “corrupção” nos locais originais.

O main() deve ficar mais ou menos assim:

```
int main(){
    <declaração e variáveis adicionais>
    char *palavra = lerDados();
    char *palavra_pre_processada = eliminarCorrupcao (palavra, <possíveis parâmetros adicionais>);
    char *palavra_criptografada = criptografaDados(palavra_pre_processada, <possíveis parâmetros
adicionais>);
    printf("A string criptografada é: %s\n", palavra_criptografada);
    char *palavra_original_pre_processada = descriptografaDados(palavra_criptografada, <possíveis
parâmetros adicionais>);
    char *palavra_original = recriarCorrupcao(palavra_original_pre_processada, <possíveis parâmetros
adicionais>);
    printf("A string original é: %s\n", palavra_original);
}
```

Exemplo de entrada de dados:

Digite a string de mensagem:

abcdefghijklmnop corrupção abacazzzztado baabxxbccd corrupção abcd (**digitado pelo usuário**)

A string criptografada é: cdebafghijlmnop zzzzabacazzzzzzztadozzzz bbbd cdab

A string original é: abcdefghijlmnop corrupção abacazzzztado baabxxbccd corrupção abcd

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

Data de Entrega: 12/05/2016.

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 !!!