Trabalho Prático 2 - Sistema de Autenticação

Em uma cifra de César, cada caractere é deslocado da sua posição um número fixo de lugares; por exemplo, considerando um deslocamento de 3, "a" torna-se "d", "b" torna-se "e", ..., "z" torna-se "a" (vamos considerar apenas caracteres minúsculos de "a" até "z"). Para ilustrar, suponha que desejamos criptografar o texto "cruzeiro". Para isso, vamos utilizar a cifra de César, considerando um deslocamento de 3 caracteres:

Entrada: `cruzeiro`
Saída: `fuxchlur`

Note que "z" + 3 = "c". Como estamos considerando apenas caracteres de "a" até "z", quando passamos do último caractere, retornamos para o "a". Isto é,

$$\begin{aligned} z + 1 &= a \\ z + 2 &= b \\ z + 3 &= c \\ &... \end{aligned}$$

A cifra de Vigenère é um método de criptografia que usa uma série de diferentes cifras de César baseadas em letras de uma "palavra-chava" (*key*). Trata-se de uma versão simplificada de uma cifra de substituição polialfabética mais geral, inventada por Leon Battista Alberti cerca de 1465. Considere, por exemplo, que a chave a ser utilizada é "key". O alfabeto possui 26 letras. Vamos considerar a letra "a" como sendo o deslocamento 0, "b" deslocamento 1, e assim por diante, até "z" como deslocamento 25. A cifragem acontece da seguinte forma: deslocamos o primeiro caractere do texto de entrada utilizando o primeiro caractere da palavra-chave (neste caso, "k", que corresponde a um deslocamento de 10); em seguida, deslocamos o próximo caractere utilizando o próximo caractere da palavra-chave (neste caso, "e", deslocamento de 4); para o terceiro caractere, novamente movemos para o terceiro caractere da chave (o caractere "y", deslocamento de 24); para o quarto caractere, já esgotamos todos os caracteres da palavra-chave, então voltamos para o primeiro (i.e. "k"). Para ilustrar, vamos criptografar o texto "cruzeiro" com a palavra-chave "key":

Entrada: `cruzeiro`
Saída: `mvsjigbs`

Note que para descriptografar a saída basta fazer o processo contrário. Isto é, ao invés deslocar o caractere para a direita o deslocamos para a esquerda (de maneira circular). Neste trabalho, vamos armazenar pares de usuários e senhas, utilizando a cifra de Vigenère para criptografar cada senha. Em seguida, iremos buscar por um determinado usuário e senha, descriptografar a senha armazenada e comparar com a senha informada na busca. É importante ressaltar que a estratégia utilizada neste exercício serve apenas para fins educativos, não sendo adotada na prática. Você deve seguir os seguintes passos:

1. Criar uma estrutura `Usuario` com campos `usuario` (string, representada como um ponteiro para caracteres) e `senha` (string);

2. Criar uma estrutura `BancoDados` que contém um campo `n` (inteiro, a quantidade de usuários) e um campo `usuarios` (um arranjo dinâmico de `n` usuários; ou seja, um ponteiro para `Usuario`).

3. Criar uma função `char *vignere(char *texto, char *chave)`, que utiliza a cifra de Vignère para criptografar o `texto` utilizando como palavra-chave o parâmetro `chave`. A função deve retornar uma string (arranjo de char) contendo o texto cifrado.

   Dica 1: Vamos trabalhar apenas com caracteres de "a" até "z". Logo, você pode subtrair de cada caractere o caractere "a" (ou seja, o código ASCII de "a"), para obter códigos de 0 a 25 ("a" = 0, "b" = 1, ...) e fazer as operações necessárias. Lembre-se que, ao final das operações, você deve somar "a" novamente para obter o código ASCII correto do caractere criptografado.

   Dica 2: Lembre-se que o deslocamento é **circular** ("z" + 1 = "a") e que a utilização da chave também é **circular** (ao esgotar todos os caracteres da chave, retornamos ao primeiro e prosseguimos com o processo de cifragem).

   Dica 3: Lembre-se que strings sempre terminam com o caractere `\0`. Logo, você deve garantir que a string retornada possua o caractere `\0` indicando o final dela.

   Dica 4: Você deverá alocar dinamicamente o arranjo a ser retornado (`malloc` ou `calloc`). Note que o tamanho do texto cifrado é o mesmo que o texto original. Você pode utilizar a função `strlen` (`string.h`) para saber o tamanho do texto original (`strlen` não inclui o `\0` no tamanho!).

4. Criar uma função `char *des_vignere(char *cifrado, char *chave)` que utiliza a cifra de Vignère para descriptografar o texto `cifrado` utilizando como palavra-chave o parâmetro `chave`. A função deve retornar uma string (arranjo de char) contendo o texto descriptografado (as dicas anteriores valem para esta função).

5. Criar uma função `autenticar(Usuario u, BancoDados bd)` que percorre o banco de dados procurando pelo usuário `u`.

   (a) Você deve fazer a busca pelo campo `u.usuario`, checando se este usuário existe no banco de dados. Se este usuário não existe, retorne falso.

   (b) Caso o usuário seja encontrado, você deve descriptografar a senha armazenada e comparar com a senha do usuário `u.senha`. Se forem iguais, retorne verdadeiro; caso contrário, retorne falso.

   Dica: Para comparar strings, você pode utilizar a função `strcmp` da biblioteca `string.h` (se o resultado for zero, as strings são iguais).

A entrada do exercício será composta de um inteiro n, seguido de n pares de usuários e senhas. Para cada par de usuário e senha, será criado um elemento do tipo `Usuario` guardando o usuario lido e a senha **criptografada**. A senha deverá ser criptografada utilizando como chave o campo `usuario`. O programa irá imprimir a senha criptografa e armazenar o `Usuario` no banco de dados. Depois, será lido um segundo inteiro m, seguido de m pares de usuários e senhas. Para cada par, será criado um `Usuario` e o programa tentará autenticar este usuário (utilizando a sua função `autenticar`). O programa irá imprimir a mensagem "Autenticação feita com sucesso!" caso a senha informada esteja correta ou "Falha na autenticação!" caso contrário. O programa principal já estará preenchido e **não deve ser alterado**. Por isso, atente-se aos nomes das funções, aos parâmetros e aos tipos de retorno. De qualquer forma, encorajamos que leia o código do programa principal e tente entender o que está sendo feito.

Exemplo de execução do programa:

Entrada	Saída
2	
luigi cruzeiro	nlcfmtlw
gleison vasco	blwkg
2	
luigi cruzeiro	Autenticação feita com sucesso!
gleison bahia	Falha na autenticação!

Para o primeiro Usuário — campos `usuario`: luigi, `senha`: cruzeiro — a chave é "luigi", o texto a ser criptografado é "senha" e a saída criptografada é "nlcfmtlw".