

ORIENTAÇÕES

Esta lista de exercícios corresponde às atividades sobre *string* e será avaliada da mesma forma que as atividades do URI.

Os exercícios abaixo devem ser resolvidos em um software adequado (CodeBlocks ou Dev) e os arquivos com a extensão `.c` devem ser enviados para o e-mail: apc.proexcelencia@gmail.com com o assunto da forma:

"String – *seu nome completo*"

Bons estudos!

Att. Equipe Pró-Excelência

Exercício 1

Você deve elaborar um programa em C que lê um nome e formata o mesmo com a primeira letra maiúscula e as demais minúsculas. O nome fornecido pelo usuário pode estar com uma combinação aleatória de letras maiúsculas e minúsculas, portanto deve ser analisado cada caractere.

Entrada: A entrada possui um nome não composto (sem espaços) e sem acentuação que pode ter de 1 a 20 caracteres com uma combinação aleatória de letras maiúsculas e minúsculas.

Saída: Deverá ser exibido na tela o nome formatado na especificação descrita: a primeira letra maiúscula e as restantes minúsculas.

Dica: Use a tabela ASCII.

Exemplo: fELisNALdO → Felisnaldo

Exercício 2

Douglas se tornou rei de todo o seu império de brinquedos e sua primeira providência consiste em criar uma língua que seja usada apenas no seu reino. No entanto, Douglas não é muito criativo e decidiu, junto ao seu conselheiro

homem-lata, que o novo idioma terá todas as frases e palavras em português invertidas.

Seus súditos tiveram muita dificuldade em escrever na nova língua e precisam de um programa que consiga traduzir as frases do português para o idioma do reino. Você é o responsável por ajudar Douglas e criar um programa em C que cumpra essa tarefa.

Entrada: A entrada possui uma frase apenas com letras minúscula, que pode conter até 50 caracteres com espaços e sem acentuação.

Saída: A saída possui a mesma frase, escrita na língua de Douglas, ou seja, com as letras e palavras invertidas.

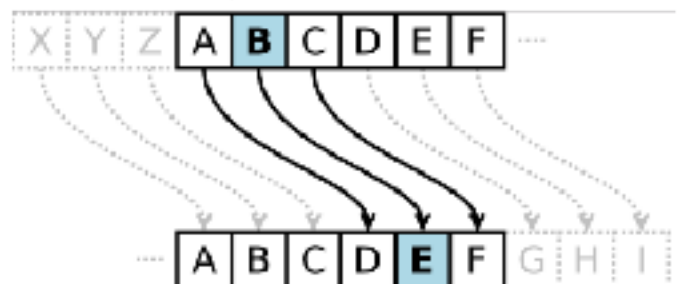
Exemplo: vida longa ao rei → ier oa agnol adv

Exercício 3

Em criptografia, temos uma técnica simples chamada de Cifra de César, que recebeu esse nome em homenagem a Júlio César. O romano usava essa técnica para codificar várias mensagens militares que destinava aos seus aliados.

A técnica consiste em substituir cada letra do texto original *segundo* um passo, que indica quantas letras serão avançadas no alfabeto. Em outras palavras, o passo define o número de letras que serão "puladas" no processo de criptografia.

No exemplo abaixo, temos o esquema de uma parte do alfabeto que foi transladada de acordo com um passo 3.



Você precisa desenvolver um programa para decodificar a frase abaixo, sabendo que a

mesma foi codificada na Cifra de César de acordo com um passo 4:

VSQE ES GSRXVEVMS I EQSV

Entrada: A entrada possui a frase codificada, apresentada no enunciado do exercício.

Saída: A saída possui a frase decodificada.

Sugestão: Fique a vontade se quiser fazer outro algoritmo para realizar o processo de codificação 😊.

Exercício 4

Alberto é um empresário muito rico que realiza várias transferências de dinheiro todos os dias. Entretanto, Alberto trabalha muito e não tem tempo suficiente para fazer todas as transferências pessoalmente e por isso confia tal tarefa a um de seus empregados.

Por segurança, Alberto codifica em palavras a quantidade de dinheiro a ser transferida ao fornecedor. Tais quantias são medidas em unidades de milhar, e tendo isso em mente, o empresário criou um código muito simples: a quantidade de letras A's em cada palavra representa o quanto, em milhares de reais, o seu empregado deve transferir.

Faça com código em C para ajudar o empregado a decodificar as palavras e saber a quantia correta a ser transferida.

Entrada: A entrada possui uma única palavra, entre 1 e 50 caracteres, sem espaços com uma determinada quantidade de letras A's em meio a uma combinação aleatória das demais letras.

Saída: A saída possui um valor, em reais, correspondente ao número de A's na palavra digitada.

Exemplo: arara → R\$ 3.000,00

Exercício 5

Você foi selecionado para desenvolver um código que cadastre no sistema uma senha e em seguida solicite a mesma, de forma que o acesso só será permitido se a senha for digitada corretamente.

O algoritmo deverá ter como critério as seguintes exigências:

- Cada senha deve ter entre 6 e 20 caracteres;
- Não deve conter espaços, acentos ou hífen;
- As senhas devem ser compostas apenas de letras minúsculas.

A senha deve ser digitada, confirmada e em seguida solicitada, para que o acesso seja permitido.

Entrada: A entrada deve ser feita em etapas: primeiro o usuário deve digitar a senha e confirmá-la logo em seguida. Em caso de erro nessa etapa, o programa deve solicitar que o usuário digite outra senha. Após a confirmação, a senha deve ser solicitada para que o login seja feito e em caso de senha incorreta, o programa deve solicitá-la novamente.

Saída: A saída deve conter o feedback de cada etapa, inclusive a de login.

Exemplo:

Digite uma senha: micalateia
Confirme sua senha: filhadeoloneia
Erro! As senhas são diferentes!
Digite uma senha: hermanoteu
Confirme sua senha: hermanoteu
Senha cadastrada com sucesso!
Digite sua senha: filhadeoloneia
Erro! Senha incorreta!
Digite sua senha: hermanoteu
Sucesso! Acesso permitido.