
	MINISTÉRIO DA EDUCAÇÃO UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS		
Curso: Sistemas de Informação		Período: 2º	Ano/Semestre: 2025.1
Disciplina: Algoritmos e Programação II		Professor: José Denes Lima Araújo	

3º ATIVIDADE – STRINGS

1. Você foi solicitado a desenvolver um programa para **criptografar mensagens** de forma que ninguém consiga entender a leitura original. O processo de criptografia é composto por **3 etapas simples**. Sua tarefa é implementar esse processo de criptografia e exibir a mensagem criptografada.

Etapas da Criptografia:

- a. **Deslocamento de +3 para as letras** (maiúsculas e minúsculas):
 - i. Para cada **letra** (maiúscula ou minúscula), aumente o valor ASCII em **3**.
 - ii. Outros caracteres (números, símbolos e espaços) **não são alterados**.
Exemplo: "a" -> "c"
- b. **Inversão da string:**
 - i. Inverter todos os caracteres da linha de entrada.
Exemplo: Se a string resultante da etapa (a) for "abc123", após a inversão ela se torna "321cba".
- c. **Deslocamento de -1 na segunda metade da string:**
 - i. Pegue todos os caracteres **da metade até o final** da string e diminua **1 no valor ASCII** de cada um.
Exemplo: se a string for de tamanho 8, subtraia 1 no valor das posições de 4 a 7 na string.

Entrada:

A entrada consiste em uma **única string** (que pode conter espaços), com até **100 caracteres**.

Saída:

Para a entrada fornecida, exiba a **mensagem criptografada** conforme o processo descrito acima.

2. Implemente um algoritmo que comprima uma string substituindo cada sequência de caracteres consecutivos iguais pelo número de repetições seguido do caractere correspondente.

Requisitos:

- i. Para cada sequência de caracteres consecutivos iguais, exiba o número de repetições seguido do caractere.
- ii. Se o caractere não se repetir, ele será representado como 1 seguido do caractere.

Exemplos:

Entrada: "AALLGGOO"

Saída: "2A2L2G2O"

Entrada: "ABCD"

Saída: "1A1B1C1D"

3. Escreva um programa em linguagem C que leia um número de CPF composto por 11 dígitos numéricos, fornecido pelo usuário (sem pontos ou hífen) e gere uma nova string com o CPF formatado no padrão: XXX.XXX.XXX-XX

Requisitos:

- **O programa deve garantir que o CPF digitado:**
 - Tenha exatamente 11 dígitos.
 - Contenha apenas dígitos numéricos (de '0' a '9').

- Caso a entrada seja inválida (quantidade incorreta de dígitos ou presença de caracteres inválidos), o programa deve exibir uma **mensagem de erro** e solicitar a entrada novamente, até que um CPF válido seja fornecido.
- O programa deve criar uma **nova string formatada**, inserindo automaticamente os **pontos (.)** e o **hífen (-)** nas posições corretas.
- A saída do programa deve exibir o CPF formatado.

Exemplo:

Entrada: "12345678900"

Saída: 123.456.789-00

4. Seu professor de português não para de trazer coisas novas para a sala, e hoje não foi diferente. Existe uma cidade, segundo seu professor, onde as pessoas levam muito a sério a forma como elas se comunica. Em especial, quando duas pessoas estão conversando, elas pensam muito nas frases antes de dizê-las, de forma a garantir que tal frase seja uma "frase completa", ou talvez uma "frase quase completa".

Considerando o nosso alfabeto de 26 letras, uma frase é dita "**completa**" quando ela contém **todas** as letras do alfabeto. De modo semelhante, uma frase é dita "**quase completa**" se ela não for completa, mas contém **ao menos metade** das letras do alfabeto. Quando uma frase não for **nem "completa" nem "quase completa"**, ela é dita "**mal elaborada**".

Seu professor lhe deu uma tarefa difícil: dada **uma** frase trocada entre dois habitantes dessa cidade, diga em qual das categorias acima ela se encaixa.

Entrada:

A entrada consiste em uma única linha contendo uma frase. A frase pode conter letras minúsculas, espaços em branco e/ou vírgulas.

Saída:

Imprima uma linha contendo uma das seguintes classificações para a frase:

- “frase completa”, quando a frase for considerada completa.
- “frase quase completa”, quando a frase for considerada quase completa.
- “frase mal elaborada”, quando a frase não for completa nem quase completa.

5. Em um reino distante, existia um velho mago chamado Eldrin, conhecido por guardar em seus pergaminhos as palavras mágicas mais poderosas do mundo. Certa vez, ele se deparou com um pergaminho antigo que continha uma palavra mágica **corrompida por duplicações de letras** — um feitiço proibido havia sido lançado por um feiticeiro invejoso.

A missão do(a) bravo(a) programador(a) é ajudar Eldrin a restaurar a palavra mágica.

Para isso, você deverá escrever um programa em C que:

1. Solicite ao usuário que **digite uma string** (palavra mágica corrompida).
2. **Verifique se há caracteres duplicados.**
3. Se houver duplicatas, **remova todos os caracteres duplicados mantendo apenas a primeira ocorrência de cada caractere.**
4. **Caso não haja caracteres duplicados, informe que "A palavra mágica está pura e sem duplicatas!"**

Exemplos de entrada e saída:

Entrada: abracadabra

Saída: Palavra restaurada: abrcd

Entrada: mundo

Saída: A palavra mágica está pura e sem duplicatas!

6. Após restaurar a palavra mágica corrompida, o velho mago **Eldrin** percebeu que a magia ainda precisava ser energizada com o **Caos das Letras** — um antigo feitiço que

embaralha a forma das letras, tornando algumas maiúsculas e outras minúsculas de maneira imprevisível.

Para isso, Eldrin precisará da sua ajuda mais uma vez!

Escreva um programa em C que:

1. **Solicite ao usuário uma string** (a nova palavra mágica).
2. **Utilize a função rand() da biblioteca stdlib.h** para gerar números aleatórios.
3. Para cada caractere da string, sorteie um número:
 - Se for par, transforme o caractere em **maiúsculo**.
 - Se for ímpar, transforme o caractere em **minúsculo**.
4. Mostre a string resultante após o feitiço do caos!

Exemplo de entrada e saída:

Entrada: eldritch

Saída: ElDrItCh (pode variar, pois é aleatório)

7. O estagiário de TI, **Lucas**, recebeu uma missão importante: revisar todas as mensagens automáticas enviadas pelo sistema da empresa. Porém, ele descobriu que algumas palavras foram digitadas com o **caractere errado!**

Para agilizar a correção, Lucas precisa de um programa que encontre e substitua todas as ocorrências de um caractere específico por outro, garantindo que as mensagens fiquem impecáveis antes do envio final.

Ajude Lucas escrevendo um programa em C que:

1. Solicite ao usuário:
 - Uma string (a mensagem original).
 - Um caractere a ser **substituído**.
 - Um caractere para **substituir**.
2. O programa deve:

- Trocar **todas as ocorrências** do primeiro caractere pelo segundo na string.
- Caso o caractere a ser substituído **não esteja presente** na string, exiba a mensagem: **"O caractere 'x' não foi encontrado na mensagem."**
- Exiba a nova mensagem após as substituições (se houverem).

8. A Agência Sombra, uma organização internacional de inteligência cibernética, interceptou uma sequência de mensagens codificadas. Cada mensagem contém uma **string encriptada**, e sua missão como analista é decodificá-la seguindo uma sequência rigorosa de transformações.

A encriptação segue um padrão específico, e **sua tarefa é reverter esse processo para revelar a mensagem original.**

1. Substitua todos os dígitos numéricos (0 a 9) pela letra 'X'.
2. Inverta a string resultante (**Exemplo: se a entrada for “abc18,.hjDF”, após passar pela etapa 1, ela será “abcXX,.hjDF”, a string resultante invertida será “FDjh,.XXcba”**).
3. **Elimine todos os sinais de pontuação** (como . , ! ? -) — somente letras devem permanecer na string final analisada
4. Por fim, verifique se a string contém uma **subsequência palindrômica de tamanho 3** (**OBS: essa subsequência é de 3 letras consecutivas**). Informe ao final se **existe ou não** tal subsequência.

Implemente um programa em C que:

- Leia uma string do usuário (máximo de 100 caracteres).
- Execute todas as etapas acima.
- Exiba a string final após as transformações.
- Indique se uma subsequência palindrômica com 3 caracteres foi detectada.

Exemplo de entrada e saída:

Entrada: 3noitamrofni ,detpyrcne si egassem!

Transformações:

1. Substituir dígitos: Xnoitamrofni ,detpyrcne si egassem!

2. Inverter: !message is encrypted ,informationX

3. Remover pontuações: message is encrypted informationX

4. Verificação de palíndromo:

Saída:

Mensagem final: message is encrypted informationX

Subsequência palindrômica de tamanho 3 não foi detectada!