

## Lista de Exercícios de Introdução à Programação II - Strings

**1ª Questão:** Faça um programa para ler uma frase de, no máximo, 80 caracteres, criptografá-la e exibir o resultado obtido. Nesse caso, a criptografia irá consistir em trocar as vogais da frase por \*.

Exemplo: Entrada: Eu estou na escola

Saída: \*\* \*st\*\* n\* \*sc\*I\*

**2ª Questão:** Faça um programa para ler uma frase de, no máximo, 100 caracteres, criptografá-la e exibir o resultado obtido. Nesse caso, a criptografia irá consistir em inverter a frase.

Exemplo: Entrada: Eu estou na escola

Saída: alocse an uotes uE

**3ª Questão:** Faça um programa para ler uma frase de, no máximo, 80 caracteres, criptografá-la e exibir o resultado obtido. Nesse caso, a criptografia irá consistir em inverter a frase e trocar as consoantes por #.

Exemplo: Entrada: Eu estou na escola

Saída: a#o##e a# uo#e# uE

**4ª Questão:** Faça um programa para ler uma frase de, no máximo, 50 caracteres, calcular e exibir a quantidade de vogais da frase digitada.

**5ª Questão:** Faça um programa para ler uma frase de, no máximo, 50 caracteres, calcular e exibir a quantidade de caracteres da frase digitada que não são vogais.

**6ª Questão:** Faça um programa para ler uma frase de, no máximo, 100 caracteres e um caracter. O programa deve calcular e exibir a quantidade de vezes que o caracter aparece na frase.

**7ª Questão:** Faça um programa para ler duas frases **F1** e **F2** de, no máximo, 50 caracteres, e gerar uma terceira **F3** que represente a concatenação das duas primeiras. Ao final, o programa deve exibir as três frases.

**8ª Questão:** Faça um programa para ler uma frase de, no máximo, 50 caracteres e uma letra. O programa deve procurar pela letra na frase e substituir todas as suas ocorrências por seu similar maiúsculo.

**9ª Questão:** Faça um programa para ler uma frase de, no máximo, 100 caracteres. O programa deve calcular e exibir quantas letras, quantos dígitos e quantos símbolos aparecem na frase.

**10ª Questão:** Faça um programa para ler uma frase de, no máximo 120 caracteres e contar e exibir quantas palavras existe na frase. Considere como delimitador de palavras o espaço em branco e os símbolos de pontuação ponto e vírgula.

**11ª Questão:** Faça um programa para checar se o dígito verificador do número de uma conta de banco está correto. O usuário deve fornecer o número da conta de banco e o seu dígito verificador. O programa deve calcular o dígito verificador do número de conta dado, comparar com o dígito fornecido pelo usuário e exibir a seguinte mensagem: "Dígito verificador válido/inválido."

**OBS1:** O número da conta deve ser uma seqüência de seis caracteres e o dígito verificador deve ser um caracter.

**OBS2:** Exemplo de como calcular o valor numérico de um caracter:

```
int num;  
char dig;  
dig = '4';  
num = dig - '0';
```

<= faz a diferença entre o valor código ascii do caracter '4' (que é 84) e o valor código ascii do caracter '0' (que é 80). **num** irá receber o valor inteiro 4.

**OBS3:** Cálculo do dígito verificador. Exemplo:

**Dados de entrada:** Conta = 123456 Dígito = 2

**Cálculos:**

Soma =  $1 \times 2^0 + 2 \times 2^1 + 3 \times 2^2 + 4 \times 2^3 + 5 \times 2^4 + 6 \times 2^5 = 321$

Resto = Soma % 11 =  $321 \% 11 = 2$

Se  $0 \leq \text{Resto} \leq 9$  então DígitoCalculado = Resto = 2

Se Resto = 10 então DígitoCalculado = X

Como, no exemplo, o dígito calculado é igual ao dígito dado, então:

**Saída:** "Dígito verificador válido".

## Funções de caracteres

Biblioteca **ctype.h**

int **isalnum** (char) : retorna um valor diferente de zero, caso o caracter seja alfanumérico, ou seja, letra (maiúscula ou minúscula) ou dígito. Caso contrário, retorna zero.

int **isalpha** (char) : retorna um valor diferente de zero, caso o caracter seja uma letra (maiúscula ou minúscula). Caso contrário, retorna zero.

int **isdigit** (char) : retorna um valor diferente de zero, caso o caracter seja um dígito. Caso contrário, retorna zero.

int **islower** (char) : retorna um valor diferente de zero, caso o caracter seja uma letra minúscula. Caso contrário, retorna zero.

int **isupper** (char) : retorna um valor diferente de zero, caso o caracter seja uma letra maiúscula. Caso contrário, retorna zero.

int **isspace** (char) : retorna um valor diferente de zero, caso o caracter seja um espaço. Caso contrário, retorna zero.

char **tolower** (char) : retorna o caracter minúsculo equivalente, se o caracter passado for uma letra maiúscula. Caso contrário, retorna o caracter inalterado.

char **toupper** (char) : retorna o caracter maiúsculo equivalente, se o caracter passado for uma letra minúscula. Caso contrário, retorna o caracter inalterado.