

M9L: Manipulação de strings

Murilo Dantas

PROGRAMA EXEMPLO 1: Faça um programa que leia uma frase de até 80 caracteres, e duas letras quaisquer do usuário. A seguir, troque na frase todas as ocorrências da primeira letra fornecida pela segunda e imprima a nova frase.

```
#include <stdio.h>

#define TAMANHO 80

void main()
{
    char frase[TAMANHO], trocar, por;
    int i;

    // Leitura da frase

    do {
        printf("Qual a frase ? ");
        gets(frase);
    } while (frase[0] == '\0');

    // Letra a ser substituída na frase

    printf("Trocar... ");
    _flushall();
    scanf("%c", &trocar);

    printf("Por... ");
    _flushall();
    scanf("%c", &por);

    // Troca das letras em toda a frase

    for(i=0; i<TAMANHO && frase[i] != '\0'; i++)
    {
        if(frase[i] == trocar)
            frase[i] = por;
    }

    printf("A nova frase eh: %s\n", frase);
}
```

1. Faça um programa que leia uma frase de no máximo 80 caracteres e imprima o tamanho real da frase lida.
2. Para enviar mensagens que não devem ser lidas por estranhos, pode-se codificá-las. Faça um programa que leia uma frase de até 100 caracteres e a seguir codifique essa frase da seguinte forma: cada letra que se encontra em posição ímpar na tabela ASCII tem este seu valor ASCII dividido por 2, e cada letra que se encontra em posição par na mesma tabela é trocada por outra, 3 posições atrás dela da tabela. No final, imprima a frase codificada.

PROGRAMA EXEMPLO 2 (INCOMPLETO) Complete o programa abaixo para que leia uma frase de até 80 caracteres e, a seguir, imprima:

- a. Quantos caracteres foram digitados.
- b. Quantos espaços brancos existem na frase.
- c. Quantos desses caracteres são minúsculos e quantos são maiúsculos.
- d. Quantos desses caracteres são dígitos.
- e. Quantos desses caracteres são de pontuação.

```
#include <stdio.h>
#include <ctype.h>

#define TAMANHO 80

void main()
{
    char frase[TAMANHO];
    int i, tamanho=0, branco=0, minusc=0, maiusc=0, digitos=0, pontuacao=0;

    // Leitura da frase

    do {
        printf("Qual a frase? ");
        gets (frase);
    } while (frase[0] == '\0');

    // Procura dos caracteres pedidos

    for(i=0; i < TAMANHO && frase[i] != '\0'; i++)
    {
        tamanho++;
        if(isspace(frase[i]))
            branco++;

        if(islower(frase[i]))
            minusc++;

        // continue aqui....

    }

    printf("A frase contem %d caracteres\n", tamanho);
    printf("O numero de espacos em branco eh de %d\n", branco);
    printf("O numero de caracteres minusculos eh de %d\n", minusc);

    // continue aqui
}
```

3. Faça um programa que leia uma frase de até 80 caracteres e inverta todas as letras maiúsculas para minúsculas e vice-versa. Além disso, o programa deve colocar um hífen no lugar de todos os espaços em branco.

PROGRAMA EXEMPLO 3: Defina uma matriz de caracteres para guardar um texto de no máximo 100 linhas de 80 caracteres cada para fazer um editor de texto que leia as várias linhas dadas pelo usuário e quando este digitar uma linha em branco, reescreva o texto do usuário e imprima as seguintes estatísticas do texto: número de caracteres digitados, número de espaços em branco e número de linhas.

```

#include <stdio.h>
#include <ctype.h>

#define LINHAS 100
#define COLUNAS 80

void main()
{
    char texto[LINHAS][COLUNAS];
    int i, j, caracteres=0, brancos=0, linhas=0;

    // Leitura do texto do usuário, com cálculo das estatísticas pedidas

    for(i=0; i<LINHAS; i++)
    {
        printf("%d : ", i+1); // Prompt para o usuário saber em que linha ele está
        for(j=0; j<COLUNAS; j++) // Nada impede a digitação além do limite!
        {
            scanf("%c", &texto[i][j]); //leitura do caracter

            if(texto[i][j] == '\n') // Se o usuário digitar <enter>,
            {
                texto[i][j] = '\0'; //coloca o caracter nulo no lugar
                break; // e encerra a leitura da linha atual.
            }

            // Estatística
            caracteres++;
            if(isspace(texto[i][j]))
                brancos++;
        }
        if(texto[i][0] == '\0') // Se o usuário digitar <enter> no começo da linha,
            break; //encerra a leitura do texto
        linhas++;
    }

    // Repetição do texto
    printf(" \nO texto digitado foi:\n");
    printf("===== \n\n");

    for(i=0; i<linhas; i++)
    {
        for(j=0; texto[i][j] != '\0'; j++)
            printf("%c", texto[i][j]);
        printf("\n");
    }

    // Impressão das estatísticas do texto
    printf("\n\n\n");
    printf("Foram digitados %d caracteres em %d linhas\n", caracteres, linhas);
    printf("Desse total, %d caracteres eram espacos em branco. \n", brancos);
}

```

4. Faça um editor de texto usando uma matriz de 25 linhas e 70 colunas que forneça as seguintes funções:
- O texto do usuário deve ser lido até que uma linha em branco seja digitada.
 - O usuário pode reimprimir seu texto digitando i.
 - O usuário pode trocar duas linhas de lugar digitando t onde o editor pergunta os números das linhas a serem trocadas entre si.

- d. O usuário pode redigitar uma linha digitando `r`, onde o editor pergunta o número da linha a ser redigitada.
- e. O usuário pode sair do editor digitando `s`.
- f. Se o usuário digitar um comando que não se encaixe em nenhum caso acima, o editor avisa que o comando não existe.