

## Lista de Exercícios 5:

### Strings:

1. Faça um programa com uma função que calcule o número de vogais existentes em uma string. A função recebe a string e atualiza a resposta (número de vogais) em uma variável que foi passada por referência.

2. Mostre e explique (comentando as linhas de código) o que será impresso na tela:

```
#include<stdio.h>
#include<string.h>
main(void)
{
    char texto[] = "foi muito facil";
    int i;
    for (i = 0; texto[i]!='\0'; i++)
    {
        if (texto[i] == ' ') break;
    }
    i++;
    for ( ; texto[i]!='\0'; i++)
    {
        printf("%c", texto[i]);
    }
}
```

3. Construa uma função que receba como argumento de entrada uma cadeia de 20 caracteres e um caractere e retorne o número de vezes que este caractere aparece na cadeia.

4. Escreva um programa em C para ler uma palavra e escrever:

-A primeira letra da palavra.

-A última letra da palavra.

-O número de letras existente na palavra (não usar a função strlen())

5. O que será impresso na tela pelo seguinte programa:

```
main(void)
{
    char frase[80]="o rato roeu a roupa do rei de roma";
    char *p;
    p = frase;
    p[2]='m';
    p[6]='\0';
    printf("%s\n", frase);
    p = p + 3;
    printf("%s\n",p);
    printf("%d\n",strlen(frase));
}
```

6. Fazer a leitura do seguinte documento:

[http://homepages.dcc.ufmg.br/~joaoreis/Site%20de%20tutoriais/c\\_int/strings.htm](http://homepages.dcc.ufmg.br/~joaoreis/Site%20de%20tutoriais/c_int/strings.htm)

7. Faça um programa que dado um nome completo, retorne a abreviatura deste nome. Não se devem abreviar as preposições como: do, de, etc. A abreviatura deve vir separada por pontos. Ex: Paulo Jose de Almeida Prado. Abreviatura: P.J.A.P.

**8.** Faça um programa que dado 2 palavras, determine:

- a. Se as palavras são iguais;
- b. Verifique se a segunda palavra é uma sub string da primeira. Ex: casa e casamento.

**9.** Faça um programa onde o usuário digita 3 informações a respeito de uma pessoa: Nome, endereço e telefone. Concatene essas três informações em uma única string.

**10.** Fazer um programa para cadastro e diálogo de login. O programa deve:

- a. Cadastrar um nome de usuário via teclado. O nome de usuário tem, no máximo, 8 caracteres, sendo válidos somente os caracteres numéricos e as letras maiúsculas ou minúsculas. Somente os caracteres válidos devem ser exibidos no console durante a digitação do nome de usuário.
- b. Cadastrar uma senha do usuário via teclado. Esta segue as mesmas regras do nome de usuário, com a diferença de que são exibidos somente asteriscos no console à medida que a senha é digitada.
- c. Receber um novo nome de usuário e uma nova senha, utilizando os mesmos procedimentos descritos nos itens a e b.
- d. Comparar o nome de usuário cadastrado com o recebido posteriormente e a senha cadastrada com a senha recebida. Caso sejam idênticos, informar “OK”, do contrário, informar “Acesso negado”.

**11.** Fazer um programa que receba uma string de no máximo 20 caracteres do usuário e mostre o conteúdo desta string de forma invertida.

**12.** Fazer um programa para receber uma string do usuário (máx. 50 caracteres) e fazer uma estatística dos caracteres digitados. Por exemplo, para a string "O EXERCICIO E FACIL", a estatística mostrada será 'O' = 2, ' '=3, 'E' = 3, 'X' = 1, 'R' = 1, 'C' = 3, 'I' = 3, 'F' = 1, 'A' = 1, 'L' = 1

**13.** Faça um programa que demonstre o uso de 3 funções da biblioteca string.h que ainda não tenham sido usadas em listas de exercícios da disciplina.