



## Aula 18 - Cadeias de caracteres

### 1 Objetivos

Apresentar e conceituar as cadeias de caracteres (strings), mostrar seu uso como vetores e mostrar funções de biblioteca para manipulação destas cadeias.

### 2 Motivação

Fazer programas que manipulem texto, por exemplo buscar palavras em um dicionário, ordenar palavras, etc.

### 3 Definição de uma cadeia de caracteres

Uma cadeia de caracteres, mais conhecida como *string*, é uma sequência de letras e símbolos, onde os símbolos podem ser espaços em branco, dígitos e vários outros como pontos de exclamação e interrogação, símbolos matemáticos, etc. Em C, uma cadeia de caracteres é representada por um vetor de variáveis do tipo `char` e é terminada com o marcador `'\0'`.

```
/* Definindo e inicializando uma cadeia de caracteres */  
char texto[100] = "Ola Mondo!";
```

```
/* Como é um vetor, posso corrigir o caractere errado da posição 5 */  
texto[5] = 'u';
```

### 4 Mostrando e lendo uma cadeia de caracteres

A sequência especial `%s` deve ser usada nos comandos `printf` e `scanf` para mostrar ou ler uma string, respectivamente.

```
/* Para mostrar na tela todo o texto armazenado na variável texto */  
printf("%s", texto);
```

```
/* Para ler um texto do teclado */  
/* (a sequência %s aplicada ao scanf lê somente uma palavra) */  
scanf("%s", texto);
```

### 5 Convertendo cadeias de caracteres em números e vice-versa

Existem variações dos comandos `printf` e `scanf` que ao invés de escrever na tela ou ler do teclado, eles escrevem/lêem de uma string. Os comandos têm um `s` (de string) na frente, portanto, são os comandos `sprintf` e `sscanf`. Estas variações funcionam de forma similar ao `printf` e `scanf`, a única diferença é que recebem como primeiro argumento a string que vai ser escrita ou lida.

```

#include <stdio.h>

int main()
{
    /* Declaramos algumas cadeias de caracteres */
    char v1[10], v2[10];

    /* Fazemos algumas contas */
    float f = 3.25 * 5.73;
    int i = 5*4*3*2*1;

    /* Conversões de números para strings */

    sprintf(v1, "%.1f", f); /* escreve o valor de f para string v1 */
    printf("%s\n", v1);     /* mostra string v1 */

    sprintf(v2, "%d", i);   /* escreve o valor de i para string v2 */
    printf("%s\n", v2);     /* mostra string v2 */

    /* Conversões de strings para números */

    float f2;
    int i2;

    sscanf(v1, "%f", &f2); /* lê um float da string v1 para f2 */
    printf("%.2f\n", f2);   /* mostra float f2 (perdeu precisão) */

    sscanf(v2, "%d", &i2); /* lê um int da string v2 para i2 */
    printf("%d\n", i2);    /* mostra int i2 */

    return 0;
}

```

## 6 Manipulando cadeias de caracteres

As cadeias de caracteres são tão importantes que existe uma biblioteca de funções só com comandos para ela. É a biblioteca **string.h**. Entre as diversas funcionalidades oferecidas por esta biblioteca, podemos destacar:

<code>strlen(texto)</code>	Retorna o tamanho da string <b>texto</b> em número de caracteres
<code>strstr(texto, busca)</code>	Procura a sub-string <b>busca</b> dentro da string <b>texto</b> e retorna um ponteiro para primeira ocorrência dela em <b>texto</b> , ou <b>NULL</b> (nulo) se não encontrar
<code>strcpy(destino, fonte)</code>	Copia a string <b>fonte</b> para a string <b>destino</b>
<code>strcat(destino, fonte)</code>	Concatena a string <b>fonte</b> no fim da string <b>destino</b>

Usamos estas operações no exemplo a seguir.

```

#include <stdio.h>
#include <string.h>

#define TAMANHO 200

```

```

int main()
{
    /* Criamos um dicionário de frutas */
    char frutas[TAMANHO] = "banana abacate laranja goiaba abacaxi";

    /* Qual porcentagem do tamanho total está sendo utilizada? */
    int em_uso = strlen(frutas);
    printf("Voce esta usando %d caracteres, ", em_uso);
    printf("ou %d%% do total.\n", em_uso*100/TAMANHO);

    /* Buscar por uma palavra (sub-string) */
    char busca[] = "goiaba";
    char *achou = strstr(frutas, busca);
    if (achou) {
        printf("Achei a palavra '%s' a partir de '%s'.\n", busca, achou);
    }

    /* Copiar e concatenar uma string */
    char str1[10] = "Ola,", str2[10] = "Mundo.";
    char texto[100];
    strcpy(texto, str1); /* copia str1 para string texto */
    strcat(texto, str2); /* concatena no fim do texto a string str2 */
    printf("%s\n", texto);
}

```

## 7 Exercícios (com respostas)

### 7.1 Achar um caractere em uma string

Existe uma função na biblioteca `string.h` chamada `strchr` que acha a primeira ocorrência de um caractere em uma string. Escreva uma implementação para esta função através da manipulação de strings como vetores.

```

char *strchr(char *s, int c)
{
    /* busca desde o começo da string */
    char *busca = s;

    /* compara cada caractere em busca do caractere pedido */
    /* tomar cuidado para parar no marcador '\0' */
    while ((*busca != '\0') && (*busca != c)) {
        busca++;
    }

    /* Testa por qual das duas condições o while parou */
    if (*busca == c)
        return busca;
    else
        return 0;
}

```

```

/* TESTE PARA FUNÇÃO STRCHR */

#include <stdio.h>

int main()
{
    char *busca;

    if (busca = strchr("papagaio", 'g'))
        printf ("Achei a partir de '%s'.\n", busca);
}

```

## 7.2 Inserindo uma palavra

Não existe em nenhuma biblioteca uma função que permita inserir uma sub-string no meio de outra string. Esta funcionalidade pode ser implementada manipulando a string como vetor de caracteres. Escreva uma implementação para a nova função **strins** que receba como parâmetro a string de destino, a string de origem, a sub-string a ser inserida e a posição na qual a sub-string vai ser inserida. O retorno desta função pode ser o tamanho da string de destino.

```

int strins(char *destino, char *fonte, char *sub, int pos)
{
    int i;

    /* copia caracteres iniciais para destino */
    for (i = 0; i < pos; i++) {
        destino[i] = fonte[i];
    }

    /* copia sub-string para o destino enquanto não chega no marcador */
    i = 0;
    while (sub[i] != '\0') {
        destino[pos + i] = sub[i];
        i++;
    }

    /* Neste ponto a variavel i armazena o tamanho da sub-string */
    int tamanho_sub = i;

    /* copia o resto dos caracteres da fonte para o destino */
    i = pos;
    while (fonte[i] != '\0') {
        destino[tamanho_sub + i] = fonte[i];
        i++;
    }

    /* retorna o tamanho da string destino */
    return tamanho_sub + i;
}

```

```
}
```

```
/* TESTE PARA FUNÇÃO STRINS */
```

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

```
int main()
{
    char fonte[] = "Testando para uma palavra na frase.";
    char sub[] = "inserir ";
    char destino[100];

    int len = strins(destino, fonte, sub, 14);
    printf("A frase ficou '%s'.\n", destino);
}
```

### 7.3 Palíndrome

Uma cadeia de caracteres é dita ser palíndrome se a sequência dos caracteres da cadeia da esquerda para direita é igual a sequência de caracteres da direita para esquerda. Por exemplo: as seguintes strings são palíndromes: ABC12321CBA, ACCA, XYZ6.6ZYX.

Faça uma função que retorna verdadeiro se a cadeia de caracteres enviada como parâmetro é palíndrome.

```
#include <string.h>
```

```
int palindrome(char *str)
{
    int palind, i, ultimo, meio;

    /* indice do último caractere em str */
    ultimo = strlen(str) - 1;

    /* indice do caractere do meio em str */
    meio = strlen(str) / 2;

    /* assumo que str é palíndrome */
    palind = 1;

    /* verifico se realmente é palíndrome */
    i = 0;
    while (palind && i <= meio) {
        palind = (str[i] == str[ultimo-i]);
        i++;
    }

    return palind;
}
```

```

/* TESTE PARA FUNÇÃO PALINDROME */

#include <stdio.h>

int main()
{
    int i;

    char texto[][50] = {
        "ABC12321CBA",
        "ACCA",
        "XYZ6.6ZYX",
        "LALALA",
        ""
    };

    for (i=0; texto[i][0]; i++) {
        printf ("O texto '%s' ", texto[i]);
        if (palindrome(texto[i]))
            printf("eh palindrome.\n");
        else
            printf("NAO eh palindrome.\n");
    }
}

```

## 8 Referências

Estas aulas foram baseadas nas notas de aula do **prof. Alexandre Falcão**

(<http://www.dcc.unicamp.br/~afalcao/mc102/notas-aula.pdf>)

e na apostila do **prof. Flávio Keidi Miyazawa**.