

Profa. Michele Fúlvia Angelo

**PGCA 028 – Tópicos Especiais em Tecnologia
Computacional I – Introdução à Programação de
Computadores
Aula 6**

Sumário

- Vetores de Caracteres (*Strings*)

Vetores de Caracteres ou *Strings*

- Já conhecemos o tipo caracter: **char**
 - Importante: caracteres devem ser colocados entre aspas simples
- Já utilizamos cadeias de caracteres:
`printf("Esta é uma mensagem!");`
- Como utilizar variáveis para cadeias de caracteres?
- Declaração:
`char texto[100];`
- Inicialização:
`char texto[100] = {'O','l','a','!','\0'};`
`char texto2[100] = "Olá Mundo!";`

Atribuindo um valor a uma posição da cadeia:
`texto[5] = 'u';`

Vetores de Caracteres ou *Strings*

- Inicialização:
`char texto[100] = {'O','l','a','!','\0'};`
- Uma string é um vetor de tipo `char` que termina com `'\0'`. Por essa razão, uma string deve conter uma posição a mais do que o número de caracteres que se deseja.
- Constantes strings são uma lista de caracteres que aparecem entre aspas, não sendo necessário colocar o `'\0'`, que é colocado pelo compilador.

```
char a[4]={'a','b','\0'};
```

```
printf("%s\n", a);  
printf("%c", a[2]);
```

```
printf("%c", a[0]);  
printf("%c", a[3]);
```

Vetores de Caracteres ou *Strings*

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

int main(){
    char nome[50];
    char mensagem[10];
    printf("qual seu nome?");
    scanf("%s",nome);
    strcpy(mensagem,"Olá, ");
    printf("%s%s",mensagem, nome);
    getch();
}
```

Funções de Entrada e Saída

- `gets(nome_matriz);`
 - É utilizada para leitura de uma string através do dispositivo padrão, até que o ENTER seja pressionado.
 - A função `gets()` não testa limites na matriz em que é chamada.

```
main() {  
    char str[80];  
    gets(str);  
    printf("%s",str);  
}
```

Funções de Entrada e Saída

- `puts(nome_do_vetor_de_caracteres);`
 - Escreve o seu argumento no dispositivo padrão de saída (vídeo), coloca um `'\n'` no final. Reconhece os códigos de barra invertida.

```
main() {  
    puts("mensagem");  
}
```

Funções: STRCPY

- strcpy (destino, fonte);
 - Copia a string fonte para o vetor de caracteres destino.

```
main(){
    char str1[80];
    char str2[80];
    strcpy(str1,"alo");
    puts(str1);
    gets(str2);
    strcpy(str1,str2);
    puts(str1);
}
```


Funções: STRLEN

- Biblioteca: *string.h*
- `size_t strlen(string)`
- Tamanho máximo vs. Tamanho atual
 - Função `strlen` retorna tamanho atual (sem `\0`)

```
char nome[50];  
strcpy(nome, "Fulano");  
printf("%d", strlen(nome));  
scanf("%s", nome);  
printf("%d", strlen(nome));
```

Funções: STRLEN

- Caracter por caracter:
nome[2] é o 3º caracter da string nome

- Escrever a string intercalando com '*':

```
strcpy(nome, "Fulano");  
for(i=0; i<strlen(nome); i++){  
    printf("%c*", nome[i]); }
```

Saído do Programa:

F*u*l*a*n*o*

- Trocar os caracteres em índices pares por '?':

```
strcpy(nome, "Fulano");  
for(i=0; i<(strlen(nome)/2); i++){  
    nome[2*i]='?';}  
puts(nome);
```

Saído do Programa:

?u?a?o

Funções: STRCMP

- strcmp (string1, string2);
 - Compara a string1 com a string2 e retorna um valor negativo (< 0) se string1 for menor que string2, 0 (zero) se string1 for igual a string2, ou um valor positivo (> 0) se string1 for maior que string2.

```
main() {  
    char s[80];  
    int result;  
    printf("Digite a senha:");  
    gets(s);  
    result=strcmp(s,"laranja");  
    if (result==0)  
        printf(" senha ok!");  
    else  
        printf("senha inválida!") ;  
}
```

Funções: STRNCPY

- `strncpy (stringdestino, stringfonte, num);`
 - Copia os *num* primeiros caracteres da *stringfonte* para a *stringdestino*.

```
main() {  
    char str1[] = "To be or not to be";  
    char str2[6];  
    strncpy (str2, str1, 5);  
    str2[5] = '\0'; ← atenção!!!  
    puts (str2);  
}
```

Funções: STRCAT

- `strcat (stringdestino, stringfonte);`
 - Concatena a string fonte no fim da string destino.
 - `stringdestino` perde seu `\0`. A cópia termina quando é copiado o `\0` de `stringfonte`.

```
main() {  
    char um[20],dois[10];  
    strcpy(um,"bom");  
    strcpy(dois," dia");  
    strcat(um,dois);  
    printf("%s\n",um);  
}
```

Exercícios

1. Faça um programa para ler um nome e mostrá-lo de trás para frente.
Exemplo: nome lido: ANDRE DA SILVA
nome mostrado: AVLIS AD ERDNA
2. Fazer um programa que aguarda uma string do teclado e mostre na tela o valor 1 se a string digitada foi "SIM" e 0 se a string digitada foi "NAO". O programa só deve retornar alguma coisa se a string digitada for "SIM" ou "NAO".
3. Se first contiver o string “ Rutherford ” e last contiver o string “Hayes”, qual será o efeito de cada uma das chamadas?
 - a) `int n = strlen (first);`
 - b) `strcpy (first, last);`
 - c) `strncpy (first, last, 3);`
 - d) `strcat (first, last);`
 - e) `strncat (first, last, 3);`