

# STRINGS (CADEIAS DE CARACTERES)

# Introdução

- Para armazenar uma cadeia de caracteres em C utiliza-se vetores, nos quais cada posição apresenta um caractere
- **Importante:** identifica-se o fim de uma cadeia por meio do caractere nulo, ou seja, por meio de ‘\0’
  - ▣ Portanto, deve-se declarar sempre o vetor com uma posição a mais para armazenar o caractere nulo
    - Observação: o ‘\0’ é armazenado automaticamente pelo compilador

# Introdução

## □ Exemplo da palavra CADEIA

char palavra[7];

Índice	0	1	2	3	4	5	6
Valor	C	A	D	E	I	A	\0
Posição Memória	863	864	865	866	867	868	869

# Introdução

- Ler 5 nomes de 10 caracteres

```
char nomes[5][10];
```

V	E	R	O	N	I	C	A	\0	
A	N	A	\0						
J	O	A	O	\0					
M	A	R	I	A	\0				
P	A	T	R	I	C	I	A	\0	

```
for (i=0; i<5; i++)  
    gets(nomes[i]);
```

Para manipular as cadeias de caracteres em C deve-se utilizar funções específicas contidas na biblioteca `<string.h>` (ou mesmo `ctype.h` e `stdlib.h`)

# Inicializações

```
char nome[ ] = {'P', 'r', 'o', 'g', 'r', 'a', 'm', 'a', '\0'};
```

```
char nome[ ] = "Programa";
```

```
char local [ ] = "c:\\exemplos\\arquivos.txt";
```

```
char texto [ ] = "curso d\'agua";
```

```
char vet1[10], vet2[10];  
strcpy(vet1, "Programa");  
strcpy(vet2, vet1);
```

gets X scanf

scanf lê até a ocorrência do primeiro espaço em branco  
gets lê até a ocorrência do primeiro ENTER (<stdio.h>)

# Funções

## ❑ COPIAR

- ▣ `strcpy(str1, str2);`
- ▣ `strncpy(str1, str2, n);` // n primeiros

## ❑ CONCATENAR

- ▣ `strcat(cadeia1, cadeia2);`
- ▣ `strncat(cadeia1, cadeia2, n);`

## ❑ COMPARAR

- ▣ `resultado = strcmp(cadeia1, cadeia2);`
  - zero = são iguais
  - $<0$  se a cadeia1 for alfabeticamente menor que a cadeia2
  - $>0$  se a cadeia1 for alfabeticamente maior que a cadeia2
    - Considera as letras maiúsculas diferentemente das minúsculas

# Funções

## □ COMPARAR (cont.)

- ▣ resultado = strcmp(cadeia1, cadeia2);

- Considera as letras maiúsculas igualmente as minúsculas

- ▣ resultado = strncmp(cadeia1, cadeia2, n);

- Compara da posição zero até a posição n ou até encontrar alguma diferença

- Considera as letras maiúsculas diferentemente das minúsculas

- ▣ resultado = strnicmp(cadeia1, cadeia2, n);

- Compara da posição zero até a posição n ou até encontrar alguma diferença

- Considera as letras maiúsculas igualmente as minúsculas

# Funções

## □ TAMANHO

- ▣ tamanho = strlen(cadeia);

## □ INVERTENDO

- ▣ strrev(str1); // inverte todos os caracteres

## □ CONVERTENDO PARA MAIÚSCULO

- ▣ cadeia[pos] = toupper(cadeia[pos]); (<ctype.h>)

- ▣ strupr(cadeia);

## □ CONVERTENDO PARA MINÚSCULO

- ▣ cadeia[pos] = tolower(cadeia[pos]); (<ctype.h>)

- ▣ strlwr(cadeia);



# Funções

- CONVERTENDO UM VALOR NUMÉRICO EM CARACTERE
  - ▣ itoa(valor numérico inteiro (int), cadeia de caracteres, base); (<stdlib.h>)
    - Base: 2 a 36
    - Retorno de até 17bytes
  - ▣ texto = ecvt(valor numérico real (double), quantidade de dígitos significativos, quantidade total de dígitos decimais, sinal); (<stdlib.h>) //fcvt ( $\neq 3^{\circ}$  parâmetro)
- CONVERTENDO CARACTERES EM VALOR NUMÉRICO
  - ▣ (int) numero = atoi(cadeia de caracteres); (<stdlib.h>)
  - ▣ (float) numero = atof(cadeia de caracteres); (<stdlib.h>)

# Exemplo

- Exemplo de utilização das funções
- Carros – Programa que preenche um “vetor” com os modelos de cinco carros (exemplos de modelo: Fusca, Gol, etc.). Carrega outro vetor com o consumo desses carros, i.e., quantos quilômetros cada um deles faz com um litro de combustível, calcula e mostra:
  - ▣ O modelo do carro mais econômico;
  - ▣ Quantos litros de combustível cada um dos carros cadastrados consome para percorrer uma distância de 1000 quilômetros.

# Exercícios

- Faça um programa que preencha:
  - ▣ Um “vetor” com oito posições contendo nomes de lojas;
  - ▣ Outro “vetor” com quatro posições contendo nomes de produtos;
  - ▣ Uma matriz com os preços de todos os produtos em cada loja.
- O programa deverá mostrar todas as relações (nome do produto – nome da loja) em que o preço não ultrapasse R\$ 120,00.

# Exercícios

- Faça um programa que receba uma frase e mostre as letras que se repetem, juntamente com o número de repetições.

Exemplo: A PROVA FOI ADIADA

A letra A apareceu 5 vezes

A letra O apareceu 2 vezes

A letra I apareceu 2 vezes

A letra D apareceu 2 vezes

# Exercícios

- Faça um programa que receba uma frase com letras minúsculas e converta a primeira letra de cada palavra da frase para maiúscula.

Exemplo: Entrada: fazer exercícios faz bem

Saída: Fazer Exercícios Faz Bem