

# GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO PROGRAMAÇÃO DE COMPUTADORES II

Prática 12: Biblioteca <string.h>



- Cada programa deve ser efetuado em um arquivo “C” próprio contendo como nome *ex1.c* para o exercício 1, *ex2.c* para o exercício 2 e assim por diante;
- Para resolução dos exercícios aloque (*malloc*) todos os ponteiros mencionados com um valor inicial conveniente a sua codificação;
- Utilize, obrigatoriamente, as funções da biblioteca <string.h> para obter o tamanho, efetuar a comparação, cópia e concatenação de strings durante a resolução dos exercícios;

1. **Faça um programa em C que solicite ao usuário três palavras e armazene-as em três variáveis do tipo ponteiro. Após isso apresente as três palavras em ordem alfabética. Exemplo:**

Primeira palavra:	Gil
Segunda palavra:	Eduardo
Terceira palavra:	Andrade

**Ordem Alfabética:**                    **Andrade – Eduardo – Gil**

2. **Faça um programa em C que solicite ao usuário três palavras e armazene-as em três variáveis do tipo ponteiro. Após isso apresente as três palavras em ordem crescente de acordo com o seu número de caracteres. Exemplo:**

Primeira palavra:	Ana	(3 letras)
Segunda palavra:	Paula	(5 letras)
Terceira palavra:	Cardoso	(7 letras)

**Ordem Crescente:**                    **Ana – Paula – Cardoso**

3. **Faça um programa em C que solicite ao usuário três palavras e armazene-as em três variáveis do tipo ponteiro. Após isso concatene essas três palavras em uma quarta variável do tipo ponteiro e apresente o seu conteúdo. Exemplo:**

Primeira palavra:	Para
Segunda palavra:	lele
Terceira palavra:	pipedo

**Concatenado:**                    **Paralelepipedo**

4. Faça um programa em C que solicite ao usuário duas palavras e armazene-as em duas variáveis do tipo ponteiro. Após isso, utilizando a função *strncmp()* indique quantas e quais letras destas duas palavras, da esquerda para direita são iguais. Exemplo:

Primeira palavra: Camaleão  
Segunda palavra: Camarão

**Letras Iguais: 4 – Cama**

5. Faça um programa em C que solicite ao usuário um valor inteiro e armazene-a em uma variável 'n', solicite também três palavras e armazene-as em três variáveis do tipo ponteiro. Após isso, utilizando a função *strncat()* concatene as primeiras 'n' letras destas palavras em uma quarta variável do tipo ponteiro e apresente o seu conteúdo. Exemplo:

Número de letras: 2  
Primeira palavra: **Barco**  
Segunda palavra: **Nariz**  
Terceira palavra: **Navio**

**Concatenado: BaNaNa**

6. **(DESAFIO)** Faça um programa em C que receba seis nomes (*strings*) utilizando e concatene-as em ordem alfabética com separador "=>" dentro de uma variável do tipo ponteiro. Utilize para comparar, concatenar e copiar *strings* as funções *strcmp()*, *strcat()* e *strcpy()*. Imprima o conteúdo do ponteiro gerado. Exemplo:

Primeiro nome: Valério  
Segundo nome: Emílio  
Terceiro nome: Wagner  
Quarto nome: Diego  
Quinto nome: Hugo  
Sexto nome: Gil

**Concatenado: Diego=>Emílio=>Gil=>Hugo=>Valério=>Wagner**

\* *Dica: utilize uma matriz de ponteiros para armazenar cada um dos nomes em suas linhas, ordenando-os dentro desta matriz.*

7. **(DESAFIO)** Faça um programa em C que receba seis nomes (*strings*) utilizando e concatene-as em ordem crescente dentro de uma variável do tipo ponteiro, utilizando como critério o número de letras desses nomes. Utilize um separado separador "=>" entre os nomes. Para comparar, concatenar e copiar *strings* use as funções *strcmp()*, *strcat()* e *strcpy()*. Imprima o conteúdo do ponteiro gerado. Exemplo:

Primeiro nome: Valério (7 letras)  
Segundo nome: Emílio (6 letras)  
Terceiro nome: Wagner (6 letras)

Quarto nome:	Diego	(5 letras)
Quinto nome:	Hugo	(4 letras)
Sexto nome:	Gil	(3 letras)

**Concatenado:**                      **Gil=>Hugo=>Diego=>Emílio=>Wagner=>Valério**

8. **(DESAFIO)** Faça um programa em C que receba três números inteiros de dois dígitos cada que devem ser armazenados em três variáveis ponteiros do tipo *char* alocadas com tamanho dois, utilize a função *gets()*. Após isso concatene-os, utilizando as funções *strcpy()* e *strcat()* num quarto ponteiro do tipo *char* alocado com tamanho seis. Agora leia o ponteiro do tipo *char* que contém a concatenação e armazene em duas variáveis inteiras os números formados pela junção dos valores pares e pelos valores ímpares. Exemplo:

Primeiro número:	32
Segundo número:	45
Terceiro Número:	92

<b>Concatenado:</b>	<b>324572</b>
<b>Pares:</b>	<b>242</b>
<b>Ímpares:</b>	<b>359</b>