**Atividade de Ponteiros**

Executada por, Rafael Kazhito Vidal Beltrão ADS - Turma C

O código também esta disponível no meu repositório do GitHub (rkazuhito)

<https://github.com/rkazuhito/Faculdade/blob/main/2o%20semestre/estrutura%20de%20dados/atividade%20ponteiros%2029-02/ponteiro%20comentado.cpp>

-> Código comentado e explicado <-

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Função para calcular o comprimento de uma string

int calcularComprimentoString(const char \*str) {

int comprimento = 0;

// Itera sobre a string até encontrar o caractere nulo '\0'

while (\*str != '\0') {

comprimento++;

str++;

}

return comprimento;

}

// Função para copiar uma string para outra

void copiarString(char \*destino, const char \*origem) {

// Itera sobre a string de origem e copia cada caractere para a string de destino

while (\*origem != '\0') {

\*destino = \*origem;

destino++;

origem++;

}

// Adiciona o caractere nulo '\0' ao final da string de destino

\*destino = '\0';

}

// Função para concatenar duas strings

void concatenarString(char \*destino, const char \*origem) {

// Move o ponteiro para o final da string de destino

while (\*destino != '\0') {

destino++;

}

// Copia os caracteres da string de origem para o final da string de destino

while (\*origem != '\0') {

\*destino = \*origem;

destino++;

origem++;

}

// Adiciona o caractere nulo '\0' ao final da string de destino

\*destino = '\0';

}

int main() {

char string1[100]; // Declara um array para armazenar a primeira string

char string2[100]; // Declara um array para armazenar a segunda string

char continuar; // Variável para verificar se o usuário deseja continuar

do {

printf("Digite a primeira string: ");

scanf("%s", string1); // Lê a primeira string do usuário

printf("Digite a segunda string: ");

scanf("%s", string2); // Lê a segunda string do usuário

// Calcula o comprimento das strings

int comprimento1 = calcularComprimentoString(string1);

int comprimento2 = calcularComprimentoString(string2);

printf("Comprimento da primeira string: %d\n", comprimento1);

printf("Comprimento da segunda string: %d\n", comprimento2);

// Alocamos memória suficiente para a string resultante

char \*string\_concatenada = (char \*)malloc(comprimento1 + comprimento2 + 1);

// Copia a string1 para a string alocada e imprime

copiarString(string\_concatenada, string1);

printf("String1 copiada: %s\n", string\_concatenada);

// Concatena a string2 à string alocada e imprime o resultado

concatenarString(string\_concatenada, string2);

printf("String1 + String2 concatenadas: %s\n", string\_concatenada);

free(string\_concatenada); // Liberamos a memória alocada

printf("Deseja inserir mais strings? (s/n): ");

scanf(" %c", &continuar); // Lê a resposta do usuário

} while (continuar == 's' || continuar == 'S');

return 0;

}

-> Execução print<-

Nesta execução foram digitadas 4 strings. Duas na primeira passagem,



Obtendo como resultado,



Na sequência a informação de continuação do programa, para informar mais duas strings



como resultado,



e assim finalizando a execução do programa.



Print completo da execução

