TRABALHO STRINGS

NOME: KAUÃ ARAUJO DE SOUZA

PASTA DO GOOGLE DRIVE ou GITHUB: Link Aqui

EXERCÍCIO 1:

****
****

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

int main() {
   char string[100], string_invertida[100];
   int tamanho, i;

   // Entrada da string
   printf("Digite uma string (até 99 caracteres): ");
   fgets(string, sizeof(string), stdin);

   // Remove o caractere de nova linha gerado pelo fgets, se existir
   string[strcspn(string, "\n")] = '\0';

   // Determina o tamanho da string
   tamanho = strlen(string);

   // Inverte a string
   for (i = 0; i < tamanho; i++) {
```

```
    string_invertida[i] = string[tamanho - 1 - i];
  }
  string_invertida[tamanho] = '\0'; // Adiciona o caractere de terminação

  // Exibe a string original e a invertida
  printf("String original: %s\n", string);
  printf("String invertida: %s\n", string_invertida);

  return 0;
}
```

****

EXERCÍCIO 14:

****

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

int main() {
  char mensagem[200], palavra[50];
  int contador, i, j, k, encontrada;

  printf("Digite mensagens e palavras para contar. Digite 'FIM' ou 'fim' para encerrar o programa.\n");

  while (1) {
    // Entrada da mensagem
```

```
printf("\nDigite uma mensagem: ");
fgets(mensagem, sizeof(mensagem), stdin);
mensagem[strcspn(mensagem, "\n")] = '\0'; // Remove o '\n'

// Verifica se a mensagem é "FIM" ou "fim"
if (strcasecmp(mensagem, "FIM") == 0) {
   printf("Programa encerrado.\n");
   break;
}

// Entrada da palavra a ser contada
printf("Digite a palavra para buscar: ");
scanf("%s", palavra);
getchar(); // Limpa o buffer do teclado

// Contar a frequência da palavra na mensagem
contador = 0;
for (i = 0; mensagem[i] != '\0'; i++) {
   encontrada = 1;
   for (j = 0; palavra[j] != '\0'; j++) {
      if (mensagem[i + j] != palavra[j]) {
         encontrada = 0;
         break;
      }
   }
   if (encontrada && (mensagem[i + j] == ' ' || mensagem[i + j] == '\0')) {
      contador++;
   }
```

```
    }

    // Exibir o resultado
    printf("A palavra '%s' aparece %d vez(es) na mensagem.\n", palavra, contador);
  }

  return 0;
}
```

****

EXERCÍCIO 17:

****

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

// Protótipo da função
void remover_espacos(char mensagem[]);

// Função para remover todos os espaços em branco de uma mensagem
void remover_espacos(char mensagem[]) {
  int i, j = 0;
  char sem_espacos[200]; // String temporária para armazenar a mensagem sem espaços

  for (i = 0; mensagem[i] != '\0'; i++) {
    if (mensagem[i] != ' ') {
      sem_espacos[j++] = mensagem[i];
```

```
        }
    }

    sem_espacos[j] = '\0'; // Finaliza a string sem espaços
    strcpy(mensagem, sem_espacos); // Copia a string sem espaços para a original
}

int main() {
    char mensagem[200];
    int i;

    printf("Digite 50 mensagens. Todas serão exibidas sem espaços em branco:\n");

    for (i = 1; i <= 50; i++) {
        // Entrada da mensagem
        printf("\nMensagem %d: ", i);
        fgets(mensagem, sizeof(mensagem), stdin);

        // Remove o caractere de nova linha gerado pelo fgets, se existir
        mensagem[strcspn(mensagem, "\n")] = '\0';

        // Remove os espaços da mensagem
        remover_espacos(mensagem);

        // Exibe a mensagem sem espaços
        printf("Mensagem sem espaços: %s\n", mensagem);
    }
```

```
    return 0;
}
```

****

EXERCÍCIO 21:

****

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

// Protótipo da função
void inverter_palavra(char palavra[]);

// Função para inverter a palavra
void inverter_palavra(char palavra[]) {
    int i, j;
    char temp;

    // Inverte os caracteres da palavra
    for (i = 0, j = strlen(palavra) - 1; i < j; i++, j--) {
        temp = palavra[i];
        palavra[i] = palavra[j];
        palavra[j] = temp;
    }
}

int main() {
    char palavra[100];
```

```
    int i;

    printf("Digite 500 palavras. Cada palavra será exibida de forma invertida:\n");

    for (i = 1; i <= 500; i++) {
        // Entrada da palavra
        printf("\nPalavra %d: ", i);
        scanf("%s", palavra);

        // Inverte a palavra
        inverter_palavra(palavra);

        // Exibe a palavra invertida
        printf("Palavra invertida: %s\n", palavra);
    }

    return 0;
}
```

****

EXERCÍCIO 22:

****

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

// Protótipo da função
```

```
void transformar_maiusculo(char mensagem[]);

// Função para transformar uma mensagem em maiúsculas
void transformar_maiusculo(char mensagem[]) {
   for (int i = 0; mensagem[i] != '\0'; i++) {
      if (mensagem[i] >= 'a' && mensagem[i] <= 'z') {
         mensagem[i] = mensagem[i] - 'a' + 'A'; // Converte para maiúscula
      }
      // Outros caracteres permanecem inalterados
   }
}

int main() {
   char mensagem[200];
   int i;

   printf("Digite 100 mensagens. Cada mensagem será convertida para
maiúsculas:\n");

   for (i = 1; i <= 100; i++) {
      // Entrada da mensagem
      printf("\nMensagem %d: ", i);
      fgets(mensagem, sizeof(mensagem), stdin);

      // Remove o caractere de nova linha gerado pelo fgets, se existir
      mensagem[strcspn(mensagem, "\n")] = '\0';

      // Converte a mensagem para maiúsculas
```

```
        transformar_maiusculo(mensagem);

        // Exibe a mensagem em maiúsculas
        printf("Mensagem em maiúsculas: %s\n", mensagem);
    }

    return 0;
}
```

****
****