INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Charqueadas

**Tecnólogo em Sistemas para Internet**
**Disciplina de Programação Estruturada – 2 Semestre**
**André Luís Del Mestre Martins**

Lista de exercícios 3 – Strings – **Prazo: combinado em aula**

**INSTRUÇÕES:**
A – Todos os arquivos compactados em formato ZIP (.zip). Inclua apenas os códigos-fonte (.c e .h), ou seja, não me envie os executáveis (.exe). O zip deve seguir o padrão: *[SEU_NOME].zip.*
B – Todas as funções solicitadas na lista de exercícios devem estar dentro de sua biblioteca. A biblioteca deve seguir o padrão: *[SEU_NOME].h* para os protótipos e *[SEU_NOME].c* para a descrição das funções.
C – Os códigos-fonte dos exercícios devem conter APENAS a função `main()` e devem seguir o padrão: *ex[NUMERO].c.* Lembre-se que são apenas 6 arquivos.
D – A lista de exercícios é entregue pelo Google Classroom.

## *PARTE 1 – ASCII e STRING*

1 – Escreva a função `void strUppercase(char*)` para converter a string com letras maiúsculas. Saída esperada:

```
String de entrada: Programacao Estruturada
String de saida: PROGRAMACAO ESTRUTURADA
```

2 – Escreva a função `int countVoyals(char*)` para contar o numero de vogais. Voce pode utilizar a funcao `void strMaiusculas(char*)` para simplificar o teste. Saída esperada:

```
Saida: "Programacao Estruturada" tem 10 vogais
```

3 – Escreva a função `void toggleString(char*)` para substituir letras maiusculas por minusculas e vice-versa em uma string. Saída esperada:

```
String de entrada: Programacao Estruturada
String de saida: pROGRAMACAO eSTRUTURADA
```

## *PARTE 2 – CHAR e STRING*

4 – Escreva a função `int findChar(char*, char)` para encontrar a primeira ocorrencia de um caracter em uma string. Saída esperada:

```
Caracter 'r' encontrado na posicao 1 da str "Programacao Estruturada"
```

5 – Escreva a função `int countChar(char*, char)` para contar o numero de ocorrencias de um caracter em uma string. Saída esperada:

```
Caracter 'r' encontrado 4 vezes na string "Programacao Estruturada"
```

6 – Escreva a função `char findHighestFrequentChar(char*)` para encontrar o caracter que aparece mais vezes em uma string. Em caso de empate, retorne apenas UM character. Saída esperada:

```
Caracter 'a' eh o caracter mais comum em "Programacao Estruturada"
```

7 – Escreva a função `void removeChar(char*, char)` para remover todas as ocorrencias de um caracter em uma string. Saída esperada:

```
Removendo 'r' de "Programacao Estruturada"
Resultado: "Pogamacao Estutuada"
```

8 – Escreva a função `void replaceChar(char*, char, char)` para substituir todas as ocorrencias de um caracter por outro caracter em uma string. Saída esperada:

```
Substituindo 'r' por 'l' em "Programacao Estruturada"
Resultado: "Ploglamacao Estlutulada"
```

9 – Escreva a função `void removeBlank(char*)` para substituir todos os espacos, tabulacoes e quebras de linha de uma string. Saída esperada:

```
 Entrada:
"h3 {
     color: red;
     text-align: left;
     font-size: 8pt
}"
 Resultado: "h3{color:red;text-align:left;font-size:8pt}"
```

### *PARTE 3 – STRING e STRING*

10 – Escreva a função `int findWord(char*, char*)` para encontrar a primeira ocorrencia de um palavra em uma string. Saída esperada:

```
 "Estruturada" encontrado na posicao 12 em "Programacao Estruturada"
 "Programa" encontrado na posicao 0 em "Programacao Estruturada"
 "Andre" encontrado na posicao -1 em "Programacao Estruturada"
```

11 – Escreva a função `int countWord(char*, char*)` para contar o numero de ocorrencias de uma palavra em uma string. Saída esperada:

```
 "Estruturada" encontrado 1 vezes na string "Programacao Estruturada"
 "te" encontrado 6 vezes na string "teste teste teste"
```

12 – Escreva a função `void removeWord(char*, char*)` para remover todas as ocorrencias de um caracter em uma string. Saída esperada:

```
 Removendo "Progr" de "Programacao Estruturada"
 Resultado: "amacao Estruturada"
```

13 – Escreva a função `void replaceWord(char*, char*, char*)` para substituir todas as ocorrencias de um caracter por outro caracter em uma string. Saída esperada:

```
 Substituindo "Program" por "Super" em "Programacao Estruturada"
 Resultado: "Superacao Estruturada"
```