

Atividades ADNP
Semana 05 :: Strings

Instruções Gerais

- Faça cada exercício em uma função distinta.
- Utilize a extensão .c e o compilador de gcc.
- Utilize o editor de sua preferência: Code Blocks, VS Code, Dev C++, etc.

IMPORTANTE: não é permitido o uso das funções da biblioteca string.h

1. Escreva uma função que devolve o tamanho de uma string (número de caracteres).
`int stringLength(char str[])`
2. Escreva uma função que conta e devolve o número de palavras em uma string. Considere que haverá somente um espaço entre as palavras.
`int countWords(char str[])`
Ex: `char s[] = "first things first, second things latter";`
`printf("%d", countWords(s)); // saída: 6`
3. Escreva uma função que conta e devolve o número de palavras em uma string. Considere que poderá haver mais de um espaço entre as palavras, bem como, no início e final da string.
`int countWordsPlus(char str[])`
Ex: `char s[] = " first things first, second things latter ";`
`printf("%d", countWordsPlus(s)); // saída: 6`
4. Escreva uma função que imprime a quantidade de letras, dígitos e caracteres especiais em uma string. Dica: Lembre-se que é possível comparar ordem dos caracteres pela tabela ASCII.
`void stringReport(char str[])`
5. Escreva uma função que recebe uma string e a converte para letras maiúsculas. Tome cuidado: a string pode já conter letras maiúsculas. Dica: tabela ASCII.
`void stringToUpper(char str[])`

Ex: `char s[] = "All your BASE are Belong to US!";`
`stringToUpper(s);`
`printf("%s", s); // saída: ALL YOUR BASE ARE BELONG TO US!`
6. Escreva uma função que recebe uma string composta de várias palavras. A função deve modificar a letra inicial de cada palavra para maiúscula e, as demais, para minúsculas. Considere que sempre haverá ao menos um espaço entre cada palavra.
`void stringCapitalize(char str[])`

Ex: `char s[] = "welCOME To COMPUTER programming!!";`
`stringCapitalize(s);`
`printf("%s", s); // saída: Welcome To Computer Programming!!`

7. Escreva uma função que converte uma string (de qualquer tamanho) para um inteiro. Utilize a notação posicional para montar o número inteiro. Ex: $2506 = 2 \times 10^3 + 5 \times 10^2 + 0 \times 10^1 + 6 \times 10^0$.

```
int stringToInt(char textNumber[])
```

```
Ex: int n = stringToInt("1024");  
    printf("%d", n); // saída: 1024 (inteiro)
```

8. Escreva uma função que informa se duas strings são iguais (devolve 1) ou diferentes (devolve 0).

```
int stringEquals(char str1[], char str2[])
```

9. Escreva uma função que compara duas strings por ordem lexicográfica (ordem alfabética). Ela deve devolver:

- a. -1, se str1 é menor que str2
- b. 0, se str1 é igual a str2
- c. 1, se str1 é maior que str2

```
int stringCompare(char str1[], char str2[])
```

10. Escreva uma função que remove os espaços que possam existir antes e depois de uma string.

```
void stringTrim(char str[])
```

```
Ex: char s[] = "    hello world    ";  
    stringTrim(s);  
    printf("%s", s); // saída: "hello world"
```

11. Escreva uma função que informa, com 1 ou 0, se uma string está contida em outra.

```
int stringContains(char str[], char sub[])
```

```
Ex: char s[] = "first things first, second things latter";  
    int check = stringContains(s, "second");  
    // neste caso, deve devolver 1, pois a string contém a palavra "second"
```