

Exercícios de Linguagem de Programação
Cursos de Engenharia de Computação
Disciplinas de APC A - 2019

Assunto: Cadeia de Caracteres - *string*

1. Construir um programa que faz a leitura de uma cadeia de caracteres (*string*), conta quantos dos caracteres digitados são letras, quantos são números e quantos são outros símbolos. Para auxiliar, se for o caso, utilize as funções abaixo:
 - a. **isupper(c)** - retorna (**Verdade**) (número diferente de zero) se o caractere contido em **c** for maiúsculo e **0 (Falso)** em caso contrário
 - b. **islower(c)** - retorna (**Verdade**) (número diferente de zero) se o caractere contido em **c** for minúsculo e **0 (Falso)** em caso contrário
 - c. **isspace(c)** - retorna (**Verdade**) (número diferente de zero) se o caractere contido em **c** for espaço e **0(Falso)** em caso contrário
 - d. **isdigit(c)** - retorna (**Verdade**) (número diferente de zero) se o caractere contido em **c** for um dígito (de 0 a 9) e **0 (Falso)** em caso contrário
 - e. **isalpha(c)** - retorna (**Verdade**) (número diferente de zero) se o caractere contido em **c** for uma letra e **0 (Falso)** em caso contrário
 - f. **isalnum(c)** - retorna (**Verdade**) (número diferente de zero) se o caractere contido em **c** for uma letra ou um dígito e **0 (Falso)** em caso contrário

Funções da biblioteca <ctype.h>

2. Construir um programa que faz a leitura de uma frase e verifica o número de espaços, dos sinais de pontuação – ponto (.) e virgula (,), a frase possui. Utilizar contadores diferentes: um para os espaços, um para os pontos e um para as virgulas. Imprimir a frase lida e o valor dos três contadores.
3. Construir um programa que faz a leitura de uma frase e conta o número de vogais que a frase possui. Imprimir a frase lida e o valor do contador.
4. Construir um programa que faz a leitura de duas cadeias de caracteres (*string*) X e Y, com tamanhos diferentes e, monta uma terceira cadeia de caracteres (*string*) Z contendo os elementos das duas *strings* lidas. Isto é, guardar primeiro os caracteres da *string* X em Z e depois os de Y. Imprimir as *strings* lidas e a calculada.

5. Construir um programa que faz a leitura de duas cadeias de caracteres (*string*) X e Y, com tamanhos diferentes e, verifica se os elementos de uma *string* são iguais ao da outra *string*. Imprimir as *strings* lidas e mensagem.
6. Construir um programa que faz a leitura de uma frase e elabora uma lista de palavras que a frase possui e ao lado dela, o número de ocorrências dessa palavra na frase. Exemplo:

Frase Lida:

“Qualquer um pode zangar-se - isso é fácil. Mas zangar-se com a pessoa certa, na medida certa, na hora certa, pelo motivo certo e da maneira certa - não é fácil.” [Aristóteles](#)

Lista:

qualquer -1	pessoa -1
um - 1	certa - 4
pode - 1	na - 2
zangar - 2	medida - 1
se - 2	hora - 1
isso - 1	pelo - 1
e - 3	motivo - 1
fácil - 2	certo - 1
mas - 1	da - 1
com - 1	maneira - 1
a -1	não - 1

1. Construir um programa que faz a leitura de uma **cadeia de caracteres (string)**, conta quantos dos caracteres digitados são letras, quantos são números e quantos são outros símbolos. Dividir o programa no seguinte conjunto de módulos:
 - **função1**: leitura de uma cadeia de caracteres (string);
 - **função2**: recebe uma cadeia de caracteres (string) e conta a quantidade de letras, números e símbolos. Devolver pela lista de parâmetros as três quantidades encontradas;
 - **função3**: imprime a string lida e os três valores calculados;
 - **função main**: define as variáveis do problema e chama as funções.

Sugestão:

utilizar as funções: `b = isdigit(c)`; verifica se o caractere `c` é um número;

`b = isalpha(c)`; verifica se o caractere `c` é uma letra

biblioteca: `<ctype.h>`

ou, utilizar o código ASCII das letras, número (intervalo de valores).

2. Construir um programa que faz a leitura de duas cadeias de caracteres (string) `X` e `Y`, com tamanhos diferentes e, monta uma terceira cadeia de caracteres (string) `Z` contendo os valores das duas strings lidas. Isto é, guardar primeiro os caracteres da string `X` em `Z` e depois os de `Y`. Imprimir as strings: lidas e a calculada. Dividir o programa no seguinte conjunto de funções:
 - **função1**: leitura das cadeias de caracteres;
 - **função2**: recebe duas cadeias de caracteres e monta a terceira como descrito acima;
 - **função3**: imprime as cadeias de caracteres lidas e a calculada;
 - **função main**: define as variáveis do problema e chama as funções.
3. Construir um programa que faz a leitura de duas cadeias de caracteres (string) `X` e `Y`, com tamanhos diferentes e, verifica se os elementos de uma string é igual ao da outra string. Se forem iguais retornar pelo return 1 senão retornar 0. Imprimir as strings lidas e mensagem se iguais ou não. Dividir o programa no seguinte conjunto de funções:
 - **função1**: leitura das cadeias de caracteres;
 - **função2**: recebe duas cadeias de caracteres e verifica se são iguais como descrito acima;
 - **função3**: imprime as cadeias de caracteres lidas e a mensagem se iguais ou não;
 - **função main**: define as variáveis do problema e chama as funções.

Fazer um programa que:

1. Faz a leitura de uma frase. A frase deve ter no máximo 500 caracteres.

2. Faz a leitura de uma palavra.
3. Verificar se a palavra digitada existe na frase. Se existir, verificar quantas vezes

ela ocorre;

1. Imprimir, a frase lida, a palavra lida e mensagem se a palavra existe ou não e,

se existir, e a quantidade de vezes.

Obs. Em geral as palavras são seguidas por vários sinais de pontuação. Para o

exercício, supor que a frase possui apenas os sinais de pontuação:

ponto-final e vírgula.