

Lista de Exercícios 7 – Vetores e Strings

Esta Lista é dividida em três partes:

- Parte I – Strings e Vetores – Entregar no CANVAS e no Verde
- Parte II – Vetores – NÃO PRECISA ENTREGAR
- Parte III – Strings – NÃO PRECISA ENTREGAR

Observações

Código para leitura de frase (*string* longa com espaços) e armazenamento em vetor de char.

```
char frase[60];  
scanf("%[^\n]", &frase); // opção para leitura sem problema com espaços  
fflush(stdin);  
printf("%s", frase);
```

Os algoritmos dos exercícios são propostos para serem implementados usando função ou procedimento. Antes de programar para cada função/procedimentos defina comentando no código:

/*

Descrição:

Lista de Parâmetros (nome e tipo de cada um):

Tipo de retorno:

*/

Informações sobre cópias

As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

PARTE I – EXERCÍCIOS STRINGS E VETORES PARA ENTREGAR NO VERDE E CANVAS

Para cada problema proposto postar as soluções no CANVAS. Para isto, compacte em único arquivo o conjunto das soluções (os arquivos com extensão .c). Além disso, todos os problemas devem ser submetidos ao sistema verde.

1. Faça uma função que receba por parâmetro uma string e retorna o número de caracteres alfanuméricos. Faça um programa que declare as devidas variáveis e acione a função (LISTA7Q1).
2. Faça um procedimento que receba e preencha um vetor com as notas de uma turma de 10 alunos. Faça um outro procedimento que receba um vetor preenchido com as notas, calcule a média da turma e conte quantos alunos obtiveram nota acima da média. Esse procedimento deve exibir a média (duas casas depois da virgula) e o resultado da contagem. Faça um programa que declare as devidas variáveis e acione os procedimentos. (LISTA7Q2).
3. Faça uma função que receba duas frases como parâmetro e coloque todas as letras em caixa alta (maiúscula). Faça um programa que declare as devidas variáveis e acione a função (LISTA7Q3).
4. O novo padrão de placas da Mercosul adotado pelos automóveis brasileiros possuem os seguintes requisitos: a) contém 7 caracteres. b) A sequência alfanumérica é intercalada: *LLLNLNN* – sendo *L* para letra e *N* para número c) todas as letras são maiúsculas. Assim sendo, deseja-se criar um sistema para validação das placas. Crie uma função booleana que receba como parâmetro uma string e verifique se a placa está no padrão. Crie um programa que receba inúmeras placas se a placa esteja correta deve imprimir "correto" ou "incorreto" e encerre o processo quando o valor da string for "fechar". Faça um programa que declare as devidas variáveis e acione a função (LISTA7Q4).
5. Faça um procedimento que leia infinitos nomes e exiba o primeiro e o último nome em ordem alfabética. Utilize como flag, a palavra "stop", para finalizar a leitura dos nomes. Faça um programa que declare as devidas variáveis e acione o procedimento (LISTA7Q5).
6. Faça um procedimento que preencha um vetor X de 10 elementos. A seguir faça uma função que receba um vetor preenchido, teste e copie todos os valores negativos deste vetor para um novo vetor (sem deixar elementos vazios entre os valores copiados), retornando este vetor como resultado. Faça um procedimento que recebe e exibe o conteúdo de um vetor. Faça um programa que faça as devidas declarações e acione os módulos para exemplificar o seu uso (LISTA7Q6).
7. Faça uma função que receba como parâmetro uma string e um caractere e, retorne o número total de ocorrência do caractere recebido como parâmetro. Atenção não deve haver diferença entre letras maiúsculas e minúsculas (LISTA7Q7).
8. Faça um procedimento que preencha 2 vetores X e Y com 10 elementos cada um (ocupando as posições de 0 a 9 em cada vetor). Faça um outro procedimento que receba dois vetores preenchidos e gera um novo vetor com os elementos desses 2 vetores intercalados de tal forma que nas posições ímpares do novo vetor estejam os elementos do primeiro vetor e nas posições pares os elementos

do segundo vetor recebido por parâmetro. Faça um procedimento que receba e exiba o conteúdo de um vetor. Faça um programa que faça as devidas declarações e acione os módulos para exemplificar o seu uso (LISTA7Q8).

9. Crie um procedimento que receba três strings e concatenação delas e exiba o resultado. Faça um programa que faça as devidas declarações e acione os módulos para exemplificar o seu uso (LISTA7Q9).

10. Faça uma função que receba uma frase de string como parâmetro e elimine, caso houver uma sequência de espaços contínuos indesejados. Faça um programa que faça as devidas declarações e acione os módulos para exemplificar o seu uso (LISTA7Q10).

11. Em uma cidade, sabe-se hipoteticamente que, em janeiro de 2023, não ocorreu temperatura inferior a 15°C, nem superior a 40°C. Faça um programa que armazene as temperaturas de cada dia de janeiro em um vetor (de 31 posições), calcule e imprima:

- A menor e a maior temperatura ocorrida (duas casas decimais depois da virgula);
- A temperatura média (duas casas decimais depois da virgula);
- O número de dias nos quais a temperatura foi inferior à temperatura média.

Faça um programa que faça as devidas declarações e acione os módulos para exemplificar o seu uso (LISTA7Q11).

12. Crie uma função que receba uma string e um vetor de inteiros com tamanho 5, em cada posição é referente a uma vogal. Essa função deve preencher o vetor de inteiro com o número de vogais "a", "e", "i", "o" e "u" em cada posição. Atenção não deve haver diferença entre letras maiúsculas e minúsculas. Ao final do programa deve exibir o número de vogais em cada posição (LISTA7Q12).

Ex: O rato roeu a corda.

Vogais	A	E	I	O	U
Quantidade	3	1	0	4	1
Posição	0	1	2	3	4

Instruções - Parte II e Parte III

Construa um algoritmo em linguagem de programação C que tenha um menu principal para acessar funções ou procedimentos que resolvam os exercícios a seguir. Após realizar uma das operações indicadas no menu, o menu deverá voltar a ser exibido para que o usuário possa escolher uma nova opção. Lembre-se de cercar opções inválidas e criar uma opção para sair do menu. Um exemplo de organização de código com menu está disponível em:

<https://replit.com/@sandrojeronimo/ExemploMenu>

Parte II – Treino - Exercícios com vetores

13. Faça um procedimento que lê 10 números informados pelo usuário e armazena em um vetor. Imprima os elementos em ordem reversa da leitura.

14. Faça um procedimento que lê 10 números informados pelo usuário e armazena em um vetor. Imprima os elementos do vetor cujo valor seja igual ao índice da posição.

15. Faça um procedimento que preenche um vetor de 10 posições com números aleatórios. Use a função *rand* da biblioteca <stdlib.h> preencher o vetor com números inteiros gerados aleatoriamente entre 0 e 10.

16. Construa um procedimento que gere 10 temperaturas aleatórias entre 18 e 40 C e armazene em um vetor. Imprima a maior, a menor e a temperatura média. Mostre também o número de vezes que a temperatura foi abaixo da média.

17. Faça um procedimento que preenche e imprime um vetor de tamanho 10 usando a seguinte regra: se a posição do vetor é múltipla de 3, deve-se armazenar 1. Caso contrário, armazene -1.

18. Construa um procedimento que leia N números entre 0 e 9 informados pelo usuário e depois armazene em cada posição *i* de um vetor de 10 posições o número de vez que o número *i* apareceu.

19. Seja um vetor ordenado com números inteiros crescentes. Construa uma função que retorne a soma dos elementos ímpares distintos (sem repetição). Por exemplo, seja o vetor com os seguintes elementos:

`int v[] = {9, 10, 10, 11, 13, 14, 14} |`

A função deverá retornar 33, que é o resultado da soma dos seguintes elementos distintos: 9 + 11 + 13.

20. Construa um procedimento que recebe um vetor e imprima os elementos cuja posição fazem parte da série de fibonacci. Por exemplo: seja o vetor com os seguintes elementos.

`int v[5] = {9, 10, 1, 3, 16}`

Como o vetor tem 5 posições (0 a 4), então o procedimento deverá imprimir os elementos das posições 0, 1, 2, 3, que são: 9, 10, 1 e 3. O elemento 16 não será impresso, pois sua posição (4) não faz parte da série de Fibonacci.

Série de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21,

21. Faça um procedimento que preenche um vetor com os primeiros *n* primeiros elementos da série de Fibonacci. Após preencher o vetor, imprima seu conteúdo. O usuário deverá definir o tamanho de *n*.

22. Construa uma função que recebe dois vetores e retorne a soma dos elementos distintos dos dois vetores. Exemplo:

V1[] = {12, 6, 3, 90, 12}
v2[] = {0, -1, 3, 9, 6, 4}

Elementos comuns: 3 e 6. Elementos distintos: 12, 90, 12, 0, -1, 9, 4. A função deverá retornar 126 (soma dos elementos distintos).

23. Construa um procedimento que preenche um vetor com o resultado da função $f(x) = 3x^3 + 2x^2 + 1x - 5$, onde X é o índice da posição do vetor. Após preencher o vetor, imprima seu conteúdo.

24. Faça um procedimento que armazena em um vetor n números digitados pelo usuário. Posteriormente, mostre o segundo menor valor do vetor.

Parte III – Treino - Exercícios com String

25. Faça um procedimento que leia uma sequência de caracteres (string) e a imprima de forma reversa (não se deve usar funções prontas para isso).

26. Construa um algoritmo que leia duas sequencias de caracteres (string) e verifique se elas são idênticas. Você deverá implementar esse algoritmo em uma função que recebe as duas strings e retorna um valor booleano.

27. Faça um algoritmo que leia uma sequência de caracter (string) e verifique se é um palíndromo. Não se deve usar funções prontas para essa operação. Esse algoritmo deverá ser implementado como função que recebe a string e retorna 1 caso seja palíndromo e -1 caso não seja.

28. Crie um procedimento que leia três nomes os imprima em ordem crescente.

29. Crie uma função que recebe duas strings (vetores de caracteres), conte e retorne quantos caracteres do primeiro vetor são presentes no segundo vetor.

Exemplo:

V1[] = "BOM DIA"

V2[] = "BOA TARDE"

Os caracteres comuns são B O D A. Portanto, a função deve retornar 4.

30. Crie uma função que recebe duas strings A e B (vetores de caracteres), e verifica se B é uma *substring* de A. Se B for uma subtring de A então a função deverá retornar a posição de A em que B se inicia, caso contrário a função deverá retornar -1. Exemplo:

A[] = "BOM DIA"

B[] = "DIA"

Nesse exemplo a função deverá retornar 4 pois essa é a posição em A que a palavra DIA se inicia.

31. Escreva um procedimento que leia 3 nomes e armazene eles em um vetor. Em seguida, apresente:

- O nome mais longo
- O nome mais curto
- Quantidade de nomes que terminam com a letra 'a'
- Quantidade total de vogais em cada nome

32. Escreva um procedimento que leia duas palavras do teclado e determina se a segunda é um anagrama da primeira. Uma palavra é um anagrama de outra se todas as letras de uma ocorrem na outra, em mesmo número, independentemente da posição. Exemplos: ROMA, MORA, ORAM, AMOR, RAMO são anagramas entre si.

Decifre o enigma

