



## Arrays (Matrizes) em C, Strings (Arrays de Char) e Structs

**ATENÇÃO:** Todos os exercícios deverão ser resolvidos em um **ÚNICO** arquivo do bloco de notas e serão enviados via **PVANet**, isto é, aceitarei apenas arquivos *“txt”* que conterão todos os exercícios do roteiro. **NÃO** aceitarei zip, odt, doc, etc; apenas txt.

### Objetivos

Praticar os conceitos de arrays de duas dimensões, strings e structs na linguagem de programação C.

### Exercícios

Os exercícios estão organizados de uma outra forma para representar uma **entrada** e **saída** de informação para cada programa. Na coluna **entrada** representa os valores que o usuário digitaria ao executar o programa, e na coluna **saída** a representação da *tela esperada* ao executar o programa.

### Matrizes

1. Codifique, compile e execute um programa na linguagem C que leia os elementos de uma matriz inteira 5x5 e imprima apenas os elementos da diagonal principal.

#### Entrada

Seu programa terá como entrada uma matriz de 5x5 posições de números inteiros e positivos.

#### Saída

A saída de seu programa deverá ser a diagonal principal do vetor de entrada, com espaçamento entre os números e sem pulo de linha.

Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
1 1	1 1 1 1 1
1 2 3 4 5 5 4 3 2 1 6 7 8 9 0 0 9 8 7 6 1 2 3 5 7	1 4 8 7 7

2. Codifique, compile e execute um programa na linguagem C que:

- (a) Leia os elementos de uma matriz inteira 4x4;
- (b) Imprima na tela a matriz criada;
- (c) Imprima a posição (linha e coluna) dos números pares.

#### Entrada

Seu programa terá como entrada uma matriz de 4x4 posições de números inteiros e positivos.

#### Saída

A saída de seu programa deverá ser a matriz de entrada e logo em seguida e posição (linha e coluna) de todos os números pares contidas nela em um printf separado por espaços entre linha e coluna e com quebra de linha no final da sentença.

Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
1 2 3 4	1 2 3 4
1 2 3 4	1 2 3 4
1 2 3 4	1 2 3 4
1 2 3 4	1 2 3 4
	0 1
	0 3
	1 1
	1 3
	2 1
	2 3
	3 1
	3 3

3. Codifique, compile e execute um programa na linguagem C que leia os elementos de uma matriz de inteiros 3 x 3. Em seguida mostre o maior elemento da matriz e o menor elemento da matriz.

#### Entrada

Seu programa terá como entrada uma matriz de 3x3 posições de números do tipo int.

#### Saída

A saída de seu programa deverá ser o maior elemento encontrado na matriz e em seguida o menor elemento encontrado na matriz, um por linha.

Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
1 2 3 1 2 3 1 2 3	3 1
4 10 5 7 2 9 11 5 3	11 2

### Strings

1) Codifique, compile e execute um programa em C que leia o nome completo do usuário e armazene em um vetor de 50 posições. Em seguida, o sistema deve exibir o tamanho da string fornecida. **(Não utilize funções prontas da biblioteca string.h)**

#### Entrada

A entrada de seu programa deve ser um nome completo que deve ser armazenado em um vetor de 50 posições.

#### Saída

Sem utilizar a biblioteca string.h, seu programa terá como saída o tamanho da string de entrada. Como mostrado abaixo:

ENTRADA	SAÍDA
Nina Simone	11

1.a) Reescreva o programa da questão anterior com a função strlen().

#### Entrada

A entrada de seu programa deve ser um nome completo que deve ser armazenado em um vetor de 50 posições

#### Saída

Utilizando a função strlen() da biblioteca string.h, seu programa terá como saída o tamanho string de entrada. Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
BB King	7

2) Codifique, compile e execute um programa em C que receba duas palavras digitadas pelo usuário e verifique se elas são iguais ou não. (Não utilize funções prontas da biblioteca string.h)

**Entrada**

A entrada de seu programa deve ser duas palavras que serão armazenadas em vetores de tamanho 20.

**Saída**

Sem utilizar a biblioteca string.h, a saída de seu programa deverá ser **Sim** caso as duas palavras sejam iguais ou **Nao** caso sejam diferentes.

Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
Casa Casebre	Nao

2.a) Reescreva o programa da questão anterior com a função strcmp().

**Entrada**

A entrada de seu programa deve ser duas palavras que serão armazenadas em dois vetores de 20 posições cada.

**Saída**

A saída deverá ser Sim caso as duas palavras sejam iguais ou Nao caso sejam diferentes, utilizando da função strcmp() da biblioteca string.h.

Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
Casa Casa	Sim

3) Codifique, compile e execute um programa em C que receba do usuário uma string qualquer. Em seguida, o programa deve imprimir a string sem suas vogais.

**Entrada**

Seu programa deverá ter como entrada uma string de tamanho 20.

**Saída**

A saída deve retornar a string sem as vogais.

Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
Arara Azul	Rr zl

4) Codifique, compile e execute um programa em C que receba do usuário uma string qualquer. Em seguida, o programa deve imprimir a quantidade de vogais e a quantidade de consoantes presentes na string lida..

#### **Entrada**

Seu programa deverá ter como entrada uma string de tamanho 50.

#### **Saída**

A saída deve retornar dois números inteiros indicando a quantidade de vogais e consoantes na string.

Como mostrado no exemplo abaixo:

<b>ENTRADA</b>	<b>SAÍDA</b>
Rio Paranaíba Minas Gerais	12 11

### **Structs**

1) Codifique, compile e execute um programa em C que crie uma estrutura Hora com os seguintes campos: hora, minuto e segundo. Em seguida defina uma variável do tipo da estrutura Hora e leia os valores para cada um dos seus campos. Ao final, imprima os valores armazenados em cada campo da estrutura na tela.

#### **Entrada**

A entrada do seu programa deve ser três valores inteiros positivos que representam a hora, o minuto e o segundo.

#### **Saída**

A saída deverá ser os valores de entrada separados por dois pontos entre a hora, os minutos e os segundos.

Como mostrado abaixo:

<b>ENTRADA</b>	<b>SAÍDA</b>
19 30 13	19:30:13

2) Codifique, compile e execute um programa em C que crie uma estrutura Endereço com os seguintes campos: nome da rua, número, cidade, estado e cep. Em seguida defina uma variável do tipo da estrutura Endereço e leia os valores para cada um dos seus campos. Ao final, imprima os valores armazenados em cada campo da estrutura na tela.

#### **Entrada**

A entrada de seu programa deve ser o nome da rua, com tamanho máximo de 20 caracteres, o número da casa, a cidade, também com tamanho máximo de 20 caracteres, o estado, de tamanho máximo de 3 caracteres e o cep como valor inteiro positivo.

### Saída

A saída deverá ser os valores de entrada impressos em uma única linha de saída e sem pulo de linha.

Como mostrado abaixo:

ENTRADA	SAÍDA
Rua das Bromélias 80 Fictícia FT 00000100	Rua das Bromélias 80 Fictícia FT 00000100

3) Codifique, compile e execute um programa em C que crie uma estrutura Pessoa com os seguintes campos: nome, sexo, peso, data de nascimento e endereço. Em seguida defina uma variável do tipo da estrutura Pessoa e leia os valores para cada um dos seus campos. Ao final, imprima os valores armazenados em cada campo da estrutura na tela. Obs: Crie uma nova estrutura para Data e utilize a estrutura Endereço implementada no exercício anterior (2).

### Entrada

A entrada de seu programa deve ser o nome da pessoa armazenado em um vetor de 20 caracteres; o sexo com tamanho máximo de 10 caracteres; o peso do tipo inteiro positivo; a data de nascimento que deve ter como entrada separada o dia, o mês e o ano e por fim, o endereço que deverá ser separado em rua, número, bairro, estado e cep.

### Saída

A saída deverá ser o nome de entrada, o sexo, o peso, a data de nascimento separada por uma barra e o endereço. Deverá conter pulo de linha no fim da sentença.

Como mostrado abaixo:

ENTRADA	SAÍDA
Albert Einstein Masculino 70 14 03 1879 Rua das Bromelias 80 Ficticia FT 00000100	Albert Einstein Masculino 70 14/03/1879 Rua das Bromelias 80 Ficticia FT 00000100

4) Codifique, compile e execute um programa em C que crie uma estrutura representando os alunos do curso de Programação. A estrutura deve conter a matrícula do aluno, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova. O programa deve executar as seguintes ações:

a- permitir ao usuário entrar com os dados de 5 alunos

b- encontrar o aluno com maior nota da primeira prova

c- encontrar o aluno com maior média geral

d- encontrar o aluno com menor média geral

e- para cada aluno armazenado na estrutura diga se ele foi aprovado ou reprovado, considerando o valor 60 para aprovação.

#### **Entrada**

A entrada do seu programa deve ser a matrícula, o nome, a nota da primeira prova do tipo float, a nota da segunda prova do tipo float e a nota da terceira prova de 5 alunos, também do tipo float.

#### **Saída**

A saída deverá ser o nome do aluno com maior nota na primeira prova com pulo de linha no final da sentença, o nome do aluno com maior média geral com pulo de linha no final da sentença, o nome do aluno com menor média geral, com pulo de linha no final da sentença e para cada um dos 5 alunos imprimir Aprovado caso tenha média geral maior ou igual a 60 ou imprimir Reprovado caso tenha uma média geral menor que 60.

Como mostrado abaixo:

<b>ENTRADA</b>	<b>SAÍDA</b>
1 Jose 30.0 10.0 100.0	Maria
2 Maria 100.0 100.0 100.0	Maria
3 Carlos 80.0 90.0 100.0	Jose
4 Bernardo 90.0 90.0 90.0	Reprovado
5 Antonia 70.0 100.0 100.0	Aprovado
	Aprovado
	Aprovado
	Aprovado