



## Arrays em C

**ATENÇÃO:** Todos os exercícios deverão ser resolvidos e serão enviados via PVANet, a partir de agora apenas o código **.c** de cada exercício.

### Objetivos

Praticar os conceitos de arrays de uma dimensão na linguagem de programação C.

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.

### 1. Professor

**1.1.** Leia um vetor inteiro de 10 posições. Escreva seus valores na saída.

#### Entrada

Seu programa deve ter como entrada um vetor de 10 posições de números inteiros positivos.

#### Saída

Seu programa deve ter como saída os valores presentes nas posições do vetor.

Como mostrado no exemplo abaixo:

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

**1.2.** Codifique, compile e execute um programa na linguagem C que leia 10 números inteiros quaisquer no intervalo de [0, 9] e armazene em um vetor.

Caso o usuário digite um valor fora do intervalo de [0, 9] o programa deve insistir até que seja fornecido um valor válido, entre 0 e 9 (**do while**).

Em seguida, calcule/exiba a quantidade de números 2, 3 e 5 presentes na sequência. (utilize a estrutura de decisão **switch**).

#### Entrada

Seu programa terá como entrada um vetor de 10 posições de números inteiros e positivos.

#### Saída

A saída de seu programa deverá ser a quantidade de números 2, 3 e 5 presentes no seu vetor de entrada, separados por um espaçamento e sem pulo de linha.

Como mostrado no exemplo abaixo:

ENTRADA	SAÍDA
3 3 3 1 1 1 2 5 0 9 2	2 3 1
2 2 2 2 2 2 2 2 2 2	10 0 0

## 2. Roteiro

**2.1.** Leia um vetor inteiro de 10 posições. Conte e escreva quantos valores pares e ímpares ele possui.

### Entrada

Seu programa deve ter como entrada um vetor de 10 posições de números inteiros positivos.

### Saída

Seu programa deve ter como saída a quantidade de valores pares e ímpares que a entrada possui.

Como mostrado no exemplo abaixo:

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

**2.2.** Faça um vetor de tamanho 50 preenchido com o seguinte valor:  $(i+5*i) \% (i+1)$ , isto é, cada elemento da posição  $i$  irá receber o valor resultado da expressão. Em seguida imprima o vetor na tela.

**2.3.** Faça um programa que leia um vetor de 15 posições para números reais e, depois, um código inteiro.

Se o código for 1, imprima o vetor na ordem direta;

Se for 2, mostre o vetor na ordem inversa.

Caso, o código for diferente de 1 e 2 escreva uma mensagem falando que o código é inválido.

#### **Entrada**

Seu programa terá como entrada um vetor de 15 posições do tipo float e um número inteiro e positivo referente à opção. (1 ou 2).

#### **Saída**

A saída de seu programa deverá ser, caso 1, o vetor será apresentado em ordem direta, do tipo float, com duas casas decimais depois da vírgula. Caso 2, será apresentado em ordem inversa do tipo float com duas casas decimais depois da vírgula; tudo com espaçamento entre os números e sem pulo de linha.

Como mostrado no exemplo abaixo:

<b>ENTRADA</b>	<b>SAÍDA</b>
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 1	1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0
1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 2	15.0 14.0 13.0 12.0 11.0 10.0 9.0 8.0 7.0 6.0 5.0 4.0 3.0 2.0 1.0

**2.4.** Leia um vetor de **char** de 10 posições. Em seguida mostre apenas as vogais lidas e depois mostre apenas as consoantes lidas.

#### **Entrada**

Seu programa deve ter como entrada um vetor de 10 posições de caracteres.

#### **Saída**

Seu programa deve ter como saída as vogais lidas pela entrada.

Como mostrado no exemplo abaixo:

<b>ENTRADA</b>	<b>SAÍDA</b>
r p a r a n a i b a	a a a i a r p r n b

Obs: Atenção ao uso do “**fflush(stdin)**” neste código.