
Programação Estruturada

Carolina Aguilar

Vetor e Vetor de struct

Aula 18 – 2020.1

Vetor

- ❑ Agregado homogêneo.
- ❑ Ocupa áreas de memória consecutivas.
- ❑ A variável declarada como sendo um vetor armazena o endereço inicial do vetor.
- ❑ Quando declaramos um vetor precisamos especificar a quantidade máxima de elementos que ele poderá armazenar.
- ❑ Fica a cargo do programador controlar a quantidade exata de elementos armazenados no vetor.
- ❑ Para acessar um elemento específico do vetor, precisamos de um índice que é um número inteiro. O primeiro índice de um vetor é sempre o número 0.

Vetor

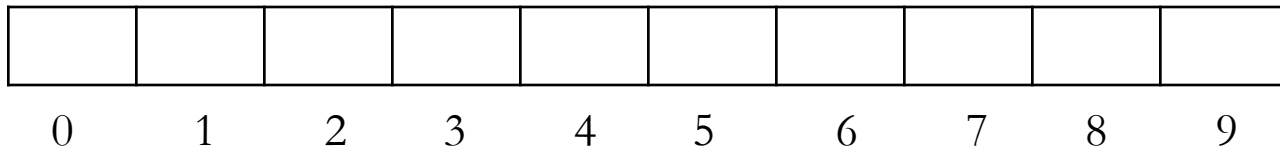
❑ Declaração:

■ Forma geral:

tipo nomeVetor[qtd];

■ Exemplo:

int vet[10];



- ❑ A variável **vet** irá armazenar o endereço inicial do vetor.
- ❑ tipo: qualquer tipo de dado necessário, incluindo tipos novos criados pelo programador.

Vetor

❑ Passando um vetor para uma função:

- Basta passar o nome da variável declarada como sendo um vetor.
- Lembre-se que na variável está guardado o endereço inicial do vetor. Passagem de parâmetro por referência. Deste modo se você modificar algum elemento do vetor dentro da função esta modificação será mantida quando a função terminar sua execução.

Vetor

- ❑ Recebendo um vetor em uma função:
 - Devemos declarar no cabeçalho da função, nos parâmetros, uma variável para armazenar o endereço inicial do vetor.
 - Exemplo: receber o vetor de inteiros **vet**, ou seja, seu endereço inicial na função **inicializa**. O tipo de retorno da função inicializa é void.

Vetor

❑ Recebendo um vetor em uma função:

■ Exemplo:

- ❑ Podemos receber de 3 formas. Independentemente da forma como a função recebe o vetor a manipulação dele dentro da função é igual.

1. `void inicializa (int *v)`
 {
 ...
 }
2. `void inicializa (int v[])`
 {
 ...
 }
3. `void inicializa (int v[10])`
 {
 ...
 }

Vetor

❑ Processando o vetor:

- Quando desejamos processar todo o vetor, devemos utilizar uma estrutura de repetição determinada, pois o programador sempre deverá controlar a quantidade exata de elementos armazenados no vetor.

■ Exemplo:

- ❑ Faça a função **inicializa** que recebe como parâmetro o vetor de inteiros **vet**, capaz de armazenar até 10 números. Esta função irá inicializar todo o vetor com 0 (zero). Não esqueça que a função recebe o endereço inicial do vetor. O tipo de retorno da função **inicializa** é void.

Vetor

❑ Processando todo o vetor:

■ Exemplo:

```
void inicializa (int *v)
{
    int cont;
    for (cont = 0; cont < 10; cont++)
        v[cont] = 0;
}
```


Vetor

□ Utilizado quando:

- Precisaríamos ler os dados de entrada mais de uma vez. Neste caso, devemos ler os dados uma única vez e armazenar em um vetor aqueles dados que iremos precisar depois.
- Precisamos de várias variáveis para um mesmo tipo de utilização. Por exemplo, quando precisamos de vários contadores, vários acumuladores etc.

Vetor de struct

- Exemplo 1: Faça um programa, utilizando a função abaixo, para processar os alunos de uma turma. Para cada aluno o programa deverá ler a matrícula e as duas notas e exibir a matrícula, a média aritmética do aluno e uma mensagem informando se a média dele está acima da média da turma, se é igual à média da turma ou se está abaixo da média da turma. Término da leitura: matrícula = 0. Considere que a turma possui, no máximo, 30 alunos. Faça a seguinte função:
 - a) `leGuarda()`: recebe como parâmetros um vetor de struct, a ser preenchido e retorna a quantidade exata de alunos armazenados no vetor de struct, ou seja, a quantidade de alunos da turma. Esta função deverá ler, para cada aluno, a matrícula e as duas notas, calcular a média do aluno, guardar no vetor de struct os dados que irá precisar depois de cada aluno. Término da leitura: matrícula = 0. A função deverá também calcular a média da turma guardando-a na variável cujo endereço é fornecido na chamada da função.

Vetor de struct

□ Exemplo 1 – Solução:

```
#include<stdio.h>
#define MAX 30
struct aluno
{
    int matr;
    float md;
};
typedef struct aluno Taluno;
```

Vetor de struct

□ Exemplo 1 – Solução (Continuação):

```
int leGuarda(Taluno v[], float *mdturma)
{
    int i, mat;
    float somamd, nota1, nota2;
    i = 0;
    somamd = 0;
    printf("Digite a matricula. 0 para encerrar: ");
    scanf("%d", &mat);
    while(mat != 0 && i < MAX)
    {
        v[i].matr = mat;
        printf("Digite as 2 notas: ");
        scanf("%f%f", &nota1, &nota2);
        v[i].md = (nota1 + nota2) / 2;
        somamd += v[i].md;
        i++;
        printf("Digite a matricula. 0 para encerrar: ");
        scanf("%d", &mat);
    }
    if(i != 0)
        *mdturma = somamd / i;
    return i;
}
```

Vetor de struct

❑ Exemplo 1 – Solução (Continuação):

```
int main()
{
    Taluno valuno[MAX];
    float mediaturma;
    int qtdalunos, cont;
    qtdalunos = leGuarda(valuno, &mediaturma);
    for(cont = 0; cont < qtdalunos; cont++)
    {
        printf("\nMatricula: %d\tMedia: %.1f\t", valuno[cont].matr, valuno[cont].md);
        if(valuno[cont].md > mediaturma)
            printf("Acima da media");
        else
            if(valuno[cont].md == mediaturma)
                printf("Igual a media");
            else
                printf("Abaixo da media");
    }
    return 0;
}
```

Vetor Contador

- Exemplo 2: Faça um programa, utilizando as funções abaixo, que leia as notas (número inteiro de 0 a 10) dos 50 alunos de uma turma e exiba a quantidade de alunos em cada nota, conforme exemplo abaixo. Faça as seguintes funções:
 - a) `inicializa()`: recebe como parâmetro um vetor de números inteiros e inicializa-o apropriadamente.
 - b) `exibe()`: recebe como parâmetro um vetor de números inteiros já preenchido e exibe a quantidade de alunos em cada nota conforme exemplo abaixo.

Exemplo:

Notas 0: 2

Notas 1: 4

...

Notas 10: 8

Vetor Contador

▣ Exemplo 2 – Solução:

```
#include<stdio.h>
#define TOT 11
void inicializa(int v[])
{
    int i;
    for(i = 0; i < TOT; i++)
        v[i] = 0;
}
void exhibe(int v[])
{
    int i;
    for(i = 0; i < TOT; i++)
        printf("\nNota %d: %d", i, v[i]);
}
```

Vetor Contador

□ Exemplo 2 – Solução (continuação):

```
int main()
{
    int vqtd[TOT], cont, nota;
    inicializa(vqtd);
    for(cont = 0; cont < 50; cont++)
    {
        printf("Digite a nota (0 a 10): ");
        scanf("%d", &nota);
        vqtd[nota]++;
    }
    exibe(vqtd);
    return 0;
}
```


Vetor Contador

❑ Observação: main () se não usássemos vetor

```
int main()
{
    int qtd0, qtd1, qtd2, qtd3, qtd4, qtd5, qtd6, qtd7, qtd8, qtd9, qtd10;
    int cont, nota;

    qtd0 = qtd1 = qtd2 = qtd3 = qtd4 = qtd5 = qtd6 = qtd7 = qtd8 = qtd9 = qtd10 = 0;

    for(cont = 0; cont < 50; cont++)
    {
        printf("Digite a nota (0 a 10): ");
        scanf("%d", &nota);
        if (nota == 0)
            qtd0++;
        else if(nota == 1)
            qtd1++;
        else if(nota == 2)
            qtd2++;
        else if(nota == 3)
            qtd3++;
        ...
        else
            qtd10++;
    }
    exhibe(qtd0, qtd1, qtd2, qtd3, qtd4, qtd5, qtd6, qtd7, qtd8, qtd9, qtd10);
    return 0;
}
```

Vetor Contador

- ❑ Exemplo 3: Faça um programa, utilizando as funções abaixo, para processar os 1000 funcionários de uma empresa. Para cada funcionário, o programa deverá ler o código do departamento no qual ele trabalha (número inteiro de 10 a 30) e exibir a quantidade de funcionários em cada departamento. Faça as seguintes funções:
 - a) `inicializa()`: recebe como parâmetro um vetor de inteiros e inicializa-o apropriadamente.
 - b) `exibe()`: recebe como parâmetro um vetor de inteiros já preenchido e exibe a quantidade de funcionários em cada departamento conforme exemplo abaixo.

Exemplo:

Departamento 10: 40

Departamento 11: 30

...

Departamento 20: 55

Vetor Contador

□ Exemplo 3 – Solução:

```
#include<stdio.h>
#define TOT 21
void inicializa(int v[])
{
    int i;
    for(i = 0; i < TOT; i++)
        v[i] = 0;
}
void exhibe(int v[])
{
    int i;
    for(i = 0; i < TOT; i++)
        printf("\nDepartamento %d: %d", i, v[i]);
}
```

Vetor Contador

□ Exemplo 3 – Solução (continuação):

```
int main()
{
    int vqtd[TOT], cont, cod;
    inicializa(vqtd);
    for(cont = 0; cont < 1000; cont++)
    {
        printf("Digite o código do departamento (10 a 30): ");
        scanf("%d", &cod);
        vqtd[cod]++;
    }
    exibe(vqtd);
    return 0;
}
```

Fim