



LABORATÓRIO DE PROGRAMAÇÃO

Estrutura de Dados em C - VETOR

Profº. Sérgio Roberto Costa Vieira, M.Sc.
Cursos de Computação
1º. Período

Laboratório de Programação

Estrutura de Dados (Vetor)

Definindo VETOR

Até o momento foi visto programas sequenciais simples, estruturas de tomada de decisão e estruturas de laço de repetição.

Estes programas utilizaram variáveis simples, espaços na memória que só podem armazenar um valor por vez durante a sua execução.

No entanto, podem existir situações que são necessárias armazenar um conjunto de valores.

Laboratório de Programação

Estrutura de Dados (Vetor)

Definindo VETOR

Vamos imaginar um problema onde precisamos armazenar um conjunto de valores...

Uma professora precisa registrar as notas de 10 alunos da turma de Matemática. Se fosse utilizar os conhecimentos até o momento adquirido, teríamos que criar dez variáveis para guardar a nota de cada aluno, ou utilizando estrutura de repetição precisaríamos apenas de uma variável, mas iríamos sobrescrever os valores das notas informadas a cada repetição.

Laboratório de Programação

Estrutura de Dados (Vetor)

Definindo VETOR

A estrutura de dados é uma técnica de programação que auxilia o programador agrupar um conjunto de valores do mesmo tipo de dados em uma única variável.

Isso significa que podemos guardar dados comuns em um único espaço da memória, essa técnica de programação é conhecida como estrutura de dados homogênea.

Nas literaturas podemos encontrar com diversos nomes: variáveis indexadas, variáveis compostas, arranjos, vetores, matrizes, tabelas em memória, arrays etc.

Laboratório de Programação

Estrutura de Dados (Vetor)

O que é VETOR?

São estruturas de dados compostas homogêneas unidimensionais que armazenam um conjunto de valores em uma única variável com o mesmo identificador (nome) e de mesmo tipo de dados (inteiro, real, caractere ou string), em vez de armazenar cada valor de forma isolada utilizando uma variável para cada valor.

Laboratório de Programação

Estrutura de Dados (Vetor)

Os valores são armazenados de forma individual, pois estes valores são indexados. O termo **indexado** refere-se a maneira de como é feita a individualização dos elementos no conjunto de dados.

Cada dado possui a sua posição e é identificado através de um índice.



Uma variável indexada pode ser definida como tendo um ou mais índices, sendo classificada como:

- Variáveis unidimensionais (vetores)
- Variáveis multidimensionais (matrizes)

Laboratório de Programação

Estrutura de Dados (Vetor)

Então o que é um VETOR?

Vetor é um espaço da memória que pode ser representado por uma linha com várias colunas consecutivas. Este espaço é identificado por um único nome, e cada coluna do vetor armazena um valor individual identificado por índices.

VetNotas									
32	17	21	8	15	29	18	54	25	12
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]

Laboratório de Programação

Estrutura de Dados (Vetor)

O que é um ÍNDICE?

O índice de um vetor é representado por um valor numérico inteiro positivo.

A quantidade de índices define o tamanho do vetor, que por vez nos permite identificar quantos valores poderão ser armazenados.

Por meio do índice também podemos identificar o conteúdo que está armazenado na estrutura de dados.

Laboratório de Programação

Estrutura de Dados (Vetor)

Declarando VETOR

As regras de declaração de vetores seguem as mesmas regras para definição de variáveis simples, mas devido ser uma variável composta é necessário definir a sua dimensão (tamanho).

Sendo que um vetor possui como características três atributos: o identificador (nome), o tipo de dados e o seu tamanho, que de modo geral utilizam-se colchetes no momento da declaração.

Laboratório de Programação

Estrutura de Dados (Vetor)

Declarando VETOR

Sintaxe: `tipo_de_dados nome_da_variável [tamanho];`

Exemplo: `int X[8];`

`float Y[7];`

– Onde:

`tipo_dados` → refere-se ao tipo de dados (char, int, float etc);

`nome_da_variável` → identifica o nome do vetor;

`[tamanho]` → é a quantidade de elementos que pode armazenar;

`int numeros[10]` → seu índice varia de 0 a 9

`char data[12]` → seu índice varia de 0 a 11

Laboratório de Programação

Estrutura de Dados (Vetor)

Atribuir valor ao VETOR

Uma vez que o vetor foi definido e reside na memória, atribuir um valor exige que seja informada em qual posição deseja-se armazenar o dado.

Lembre-se: sempre a primeira posição de um vetor tem o índice inicial de valor 0 (zero).

`vet [0] = 1 ;`

`x [3] = 'b' ;`

`sr [4] = 6.8 ;`

Laboratório de Programação

Estrutura de Dados (Vetor)

Inicialização de VETOR

Normalmente, para seu preenchimento, utiliza-se estrutura de repetição para tal tarefa.

Mas, também é possível inicializar o vetor no momento de sua declaração.

```
int idade[6] = {32, 16, 10, 35, 5, 18};
```

```
float notas[4] = {5.5, 9.0, 6.3, 8.5};
```

```
char nome[5] = "maria";
```

Laboratório de Programação

Estrutura de Dados (Vetor)

Preenchendo VETOR

A leitura de dados de um vetor significa realizar o preenchimento do vetor, elemento por elemento, conforme o usuário vai informando, o que muda apenas é o índice de posição, indicando em qual posição será armazenado cada elemento.

```
float notas[4];
```

```
for ( i = 0; i < 4; i++ ) { //Não esqueça que o índice do vetor inicia em 0 (zero)  
    printf(" \n Digite a nota do aluno: ");  
    scanf(" %f ", &notas[ i ]); //a variável i indica o índice do vetor  
}
```


Laboratório de Programação

Estrutura de Dados (Vetor)

Mostrando valores do VETOR

A saída de dados de um vetor significa mostrar os valores contidos dentro do vetor, elemento por elemento, percorrendo cada posição do vetor por meio também de um laço de repetição.

```
for ( i = 0; i < 4; i++ ) { //Não esqueça que o índice do vetor inicia em 0 (zero)
    printf(“ \n Nota: %.2f ” , notas[ i ]);
}
```

Laboratório de Programação

Estrutura de Dados (Vetor) – Exemplo 1

```
1  #include <stdio.h>
2
3  int main(){
4      int i; //variável controladora ou contadora
5
6      float v[6] = {2.3, 5.4, 1.0, 7.6, 8.8, 3.9}; //inicializando o vetor
7
8      for (i=0; i<6; i++) { //percorrendo cada indice de posição
9          .....
10         printf("%.2f ", v[i]); //imprimindo os valores
11     }
12
13
14     return 0;
15 }
```

Laboratório de Programação

Estrutura de Dados (Vetor) – Exemplo 2

```
1  #include <stdio.h>
2
3  int main(){
4
5      int i;
6      float v[6], maior;
7      //lendo os valores informados pelo usuário
8      for (i=0; i<6; i++){
9          printf("Digite o %d. valor ", i+1);
10         scanf("%f", &v[i]);
11     }
12     maior = v[0]; //incializando a variável
13     //comparando quem é o maior
14     for (i=0; i<6; i++) {
15         if(v[i] > maior){
16             maior = v[i];
17         }
18     }
19     printf("\n");
20     //imprimindo os valores informados
21     for (i=0; i<6; i++) {
22         printf("%.2f ", v[i]);
23     }
24
25     printf("\n\nO maior valor: %.2f", maior);
26     return 0;
27 }
```

Laboratório de Programação

Estrutura de Dados (Vetor) – Exemplo 3

```
1  #include<stdio.h>
2  #define NUM_ALUNOS 6
3
4  int main() {
5
6      float notas[NUM_ALUNOS], media, soma = 0.0;
7      int i;
8
9      for(i=0; i < NUM_ALUNOS; i++) {
10         printf("Entre com a nota do aluno %d: ", i+1);
11         scanf("%f", &notas[i]);
12     }
13
14     for(i=0; i < NUM_ALUNOS; i++){
15         soma = soma + notas[i];
16     }
17
18     media = soma/NUM_ALUNOS;
19
20     printf("\n Media da turma = %.2f\n", media);
21     return 0;
22 }
```

Laboratório de Programação

Estrutura de Dados (Vetor) – Exemplo 4

```
1  #include<stdio.h>
2  int main(){
3      int vet[8], i, j, aux;
4
5      for(i=0; i < 8; i++) {
6          printf("Entre com %d. valor: ", i+1);
7          scanf("%d", &vet[i]);
8      }
9      printf("\n Valores Originais \n");
10     for(i=0; i < 8; i++) {
11         printf("%d ", vet[i]);
12     }
13     for(i=0; i < 8; i++) {
14         for(j=0; j < 8; j++) {
15             if(vet[i] < vet[j]){
16                 aux = vet[i];
17                 vet[i] = vet[j];
18                 vet[j] = aux;
19             }
20         }
21     }
22     printf("\n Valores Ordenados \n");
23     for(i=0; i < 8; i++) {
24         printf("%d ", vet[i]);
25     }
26     return 0;
27 }
```


Laboratório de Programação

Exercícios de Vetores

- 1 – Crie um programa que leia do teclado seis valores inteiros e armazene em um vetor, em seguida mostre na tela os valores lidos na ordem inversa.
- 2 – Crie um programa que leia um vetor de 10 posições. Escreva na tela quantos valores pares foram armazenados nesse vetor.
- 3 – Faça um programa que preencha um vetor com 10 elementos inteiros. Calcule e mostre apenas os números ímpares e qual a quantidade de números ímpares.
- 4 – Faça um programa que receba do usuário um vetor X com 10 posições. Em seguida, deverão ser impressos o maior e o menor elemento desse vetor.

Laboratório de Programação

Exercícios de Vetores

- 5 – Crie um programa que leia um vetor com 8 posições. Em seguida mostre na tela somente os valores que se encontram nas posições pares do vetor (considere a posição zero).
- 6 – Crie um programa que leia um vetor de 10 posições. Escreva na tela quais são as posições do vetor que se encontram com valores ímpares.
- 7 – Faça um programa que preencha um vetor com 10 elementos inteiros. Calcule e mostre a média dos valores do vetor.
- 8 – Faça um programa que leia um vetor VET de 10 posições. Mostre somente os números divisíveis por 2, a sua quantidade e quantos não são divisíveis por 2.



LABORATÓRIO DE PROGRAMAÇÃO

Estrutura de Dados em C - VETOR

Profº. Sérgio Roberto Costa Vieira, M.Sc.
Cursos de Computação
1º. Período