

Fundamentos de Programação

Aula 13 - Estruturas de Dados Básicas: Vetor

Prof^a. Elisa de Cássia Silva Rodrigues

Estrutura de Dados

- Uma **estrutura de dados** representa uma forma eficiente de **armazenar e organizar dados** de modo que possam ser usados de forma eficiente.
- Uma estrutura de dados armazena **dados do mesmo tipo**.
- A utilização de estrutura de dados é importante pois a organização eficiente dos dados **diminui o custo de um algoritmo** com relação ao tempo de execução e consumo de memória.
- Estruturas de dados básicas:
 - ▶ **Vetores**.
 - ▶ **Matrizes**.
 - ▶ **Registros**.

Definição de Vetor

- **Vetor** é um conjunto de variáveis de mesmo tipo alocados **sequencialmente** na memória.
- Esse conjunto de variáveis possui o **mesmo identificador** (nome).
- Também conhecido como variável composta homogênea **unidimensional**.
- As variáveis se distinguem pelo **índice** que referencia sua **localização no vetor**.

Declaração de Vetor

Pseudocódigo

DECLARE nome[tamanho] tipo

onde:

nome é o nome da variável do tipo vetor.

tamanho é a quantidade de variáveis que vão compor o vetor.

tipo é o tipo básico dos dados que serão armazenados no vetor.

Exemplo

DECLARE X[5] inteiro

// vetor chamado X que possui cinco posições sequenciais na memória.



Declaração de Vetor

Linguagem C

```
tipo nome[tamanho];
```

onde:

nome é o nome da variável do tipo vetor.

tamanho é a quantidade de variáveis que vão compor o vetor.

tipo é o tipo básico dos dados que serão armazenados no vetor.

Exemplo

```
int X[5];
```

// vetor chamado X que possui cinco posições sequenciais na memória.



Preenchimento do Vetor

- Preencher um vetor significa atribuir valores a todas as posições.
- Para isso usa-se um laço de repetição (PARA, por exemplo).
- A variável contadora i é o índice do vetor $X[i]$.

Preenchendo um vetor X de tamanho 5 (Pseudocódigo)

```
PARA  $i \leftarrow 0$  ATÉ 4 FAÇA {  
    ESCREVA "Digite o número da posição ",  $i$ , ": "  
    LEIA  $X[i]$   
}
```

Preenchendo um vetor X de tamanho 5 (Linguagem C)

```
for( $i = 0$ ;  $i \leq 4$ ;  $i++$ ) {  
    printf("Digite o número da posição %d: ",  $i$ );  
    scanf("%d", & $X[i]$ );  
}
```

Preenchimento do Vetor

- O laço de repetição deve garantir que a variável i assuma todas as posições entre 0 e 4 (posições válidas para o vetor de tamanho 5).
- Em cada iteração do laço uma posição do vetor será preenchida.

Simulação da execução do algoritmo

Índice	Tela	Memória										
		X <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>						0	1	2	3	4
0	1	2	3	4								
i = 0	Digite o número da posição 0: 10	X <table><tr><td>10</td><td></td><td></td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	10					0	1	2	3	4
10												
0	1	2	3	4								
i = 1	Digite o número da posição 1: 9	X <table><tr><td>10</td><td>9</td><td></td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	10	9				0	1	2	3	4
10	9											
0	1	2	3	4								
i = 2	Digite o número da posição 2: 13	X <table><tr><td>10</td><td>9</td><td>13</td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	10	9	13			0	1	2	3	4
10	9	13										
0	1	2	3	4								
i = 3	Digite o número da posição 3: 7	X <table><tr><td>10</td><td>9</td><td>13</td><td>7</td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	10	9	13	7		0	1	2	3	4
10	9	13	7									
0	1	2	3	4								
i = 4	Digite o número da posição 4: 1	X <table><tr><td>10</td><td>9</td><td>13</td><td>7</td><td>1</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	10	9	13	7	1	0	1	2	3	4
10	9	13	7	1								
0	1	2	3	4								

Impressão na Tela dos Elementos do Vetor

- Segue a mesma ideia do preenchimento do vetor usando um laço de repetição.

Imprimindo um vetor X de tamanho 5 (Pseudocódigo)

```
PARA i ← 0 ATÉ 4 FAÇA {  
    ESCREVA "Número da posição ", i, ": "  
    ESCREVA X[i]  
}
```

Imprimindo um vetor X de tamanho 5 (Linguagem C)

```
for(i = 0; i <= 4; i++) {  
    printf("Número da posição %d: ", i);  
    printf("%d", X[i]);  
}
```


Exemplo:

Faça um programa que preencha um vetor com nove números inteiros, calcule e mostre os números primos e suas respectivas posições. (ASCÊNCIO, 2012)).

```
int X[9], qtd_divisores;

// preenche o vetor com nove números inteiros
for(int i=0; i<9; i++)
{
    printf("Digite X[%d]: ", i);
    scanf("%d", &X[i]);
}

printf("\nOs números primos são: \n");

//percorre o vetor X
for(int i=0; i<9; i++)
{
    qtd_divisores = 0;

    // percorre todos os números menores que X[i] e maiores que zero
    for(int j=1; j<=X[i]; j++)
    {
        // verifica se o j é divisor do número X[i]
        if(X[i] % j == 0)
        {
            qtd_divisores++;
        }
    }

    // verifica se o número X[i] tem apenas dois divisores (por ele mesmo e por 1 - primo)
    if(qtd_divisores <= 2)
    {
        // imprime o número X[i] se ele for primo
        printf("X[%d] = %d\n", i, X[i]);
    }
}
```

Sugestões de leitura:

- ▶ Capítulo 6 (Vetor) do livro texto (ASCÊNCIO, 2012).
- ▶ Capítulo 6 (Vetores e Matrizes) do livro (BACKES, 2013).

Sugestões de exercícios:

- ▶ Exercícios do capítulo 6 do livro texto (ASCÊNCIO, 2012).
 - ★ Exercícios Resolvidos: 4, 5, 9, 13, 18 e 23.
- ▶ Exercícios do capítulo 6 do livro (BACKES, 2013).
 - ★ Exercícios: 1, 2, 3 e 4.
- ▶ Lista de Exercícios 6 (Estruturas de Dados Básicas: Vetor).

- ① ASCÊNCIO, A. F. G.; CAMPOS, E. A. V. ***Fundamentos da Programação de Computadores***. 2012.
- ② BACKES, A. ***Linguagem C: completa e descomplicada***. 2013.
- ③ PAES, R. B. ***Introdução à Programação com a Linguagem C: Aprenda a resolver problemas com uma abordagem prática***. 2016.

Obrigada pela atenção!