

LPG0002 – Linguagem de Programação

Vetores

Profª Luciana Rita Guedes
Departamento de Ciência da Computação
UDESC / Joinville

Material elaborado por: Prof. Rui Jorge Tramontin Junior

Introdução

- Vetores são variáveis *multivaloradas*;

Introdução

- Vetores são variáveis *multivaloradas*;
- São variáveis *homogêneas*, ou seja, todos os valores têm mesmo tipo;

Introdução

- Vetores são variáveis *multivaloradas*;
- São variáveis *homogêneas*, ou seja, todos os valores têm mesmo tipo;
- Um vetor é sempre alocado na memória como um bloco único, e seus valores são adjacentes antes si;

Introdução

- Vetores são variáveis *multivaloradas*;
- São variáveis *homogêneas*, ou seja, todos os valores têm mesmo tipo;
- Um vetor é sempre alocado na memória como um bloco único, e seus valores são adjacentes antes si;
- Vetores têm uma capacidade definida na declaração.

Introdução

- Por exemplo, v é um vetor *int* de capacidade 10:

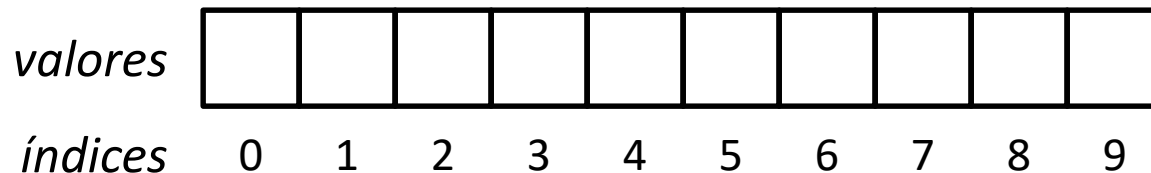
```
int v[10];
```

Introdução

- Por exemplo, v é um vetor *int* de capacidade 10:

```
int v[10];
```

- Pode-se visualizar um vetor v de maneira simples conforme o seguinte modelo:

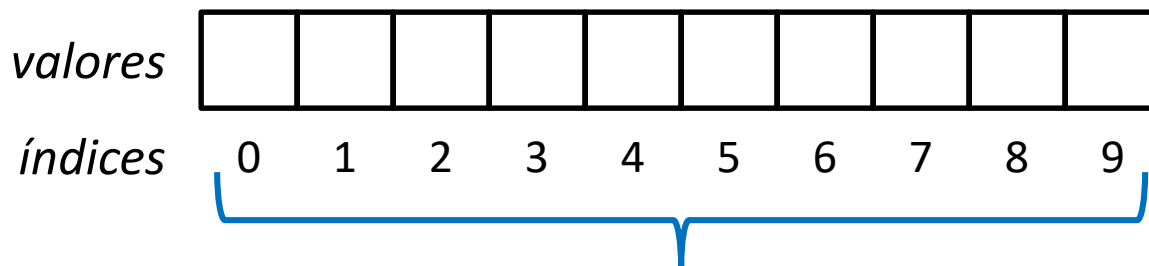


Introdução

- Por exemplo, v é um vetor *int* de capacidade 10:

```
int v[10];
```

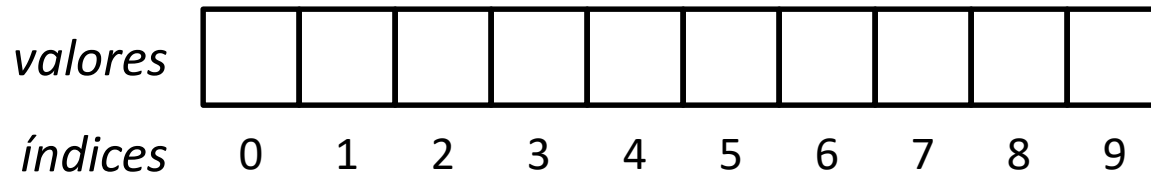
- Pode-se visualizar um vetor v de maneira simples conforme o seguinte modelo:



- Índices variam de 0 até *capacidade-1*;
- Índices permitem o acesso aos valores do vetor.

Exemplo: acesso ao vetor usando índices (1/5)

```
int v[10];
```



Exemplo: acesso ao vetor usando índices (2/5)

```
int v[10];
```

```
v[0] = -1; // Atribui -1 ao índice 0
```

<i>valores</i>	<div><div>-1</div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
<i>índices</i>	<div>0123456789</div>

Exemplo: acesso ao vetor usando índices (3/5)

```
int v[10];
```

```
v[0] = -1; // Atribui -1 ao índice 0
```

```
v[2] = 0;  // Atribui 0 ao índice 2
```

<i>valores</i>	-1		0							
<i>índices</i>	0	1	2	3	4	5	6	7	8	9

Exemplo: acesso ao vetor usando índices (4/5)

```
int v[10];
```

```
v[0] = -1; // Atribui -1 ao índice 0
```

```
v[2] = 0;  // Atribui 0 ao índice 2
```

```
v[7] = 12; // Atribui 12 ao índice 7
```

<i>valores</i>	-1		0					12		
<i>índices</i>	0	1	2	3	4	5	6	7	8	9

Exemplo: acesso ao vetor usando índices (5/5)

```
int v[10];
```

```
v[0] = -1; // Atribui -1 ao índice 0
```

```
v[2] = 0;  // Atribui 0 ao índice 2
```

```
v[7] = 12; // Atribui 12 ao índice 7
```

```
int x = 3;
```

```
v[9] = x + 1; // Atribui 4 ao índice 9
```

<i>valores</i>	-1		0					12		4
<i>índices</i>	0	1	2	3	4	5	6	7	8	9

Considerações sobre o Exemplo

- Não é a forma como os índices são utilizados na prática!

Considerações sobre o Exemplo

- Não é a forma como os índices são utilizados na prática!
- Um vetor pode ter centenas ou milhares de valores:
 - O acesso individual a cada índice torna-se inviável;

Considerações sobre o Exemplo

- Não é a forma como os índices são utilizados na prática!
- Um vetor pode ter centenas ou milhares de valores:
 - O acesso individual a cada índice torna-se inviável;
- Na prática, os valores dos índices são armazenados em uma variável, percorrendo de 0 até *capacidade-1*;

Considerações sobre o Exemplo

- Não é a forma como os índices são utilizados na prática!
- Um vetor pode ter centenas ou milhares de valores:
 - O acesso individual a cada índice torna-se inviável;
- Na prática, os valores dos índices são armazenados em uma variável, percorrendo de 0 até *capacidade-1*;
- Com isso, é possível fazer qualquer tipo de manipulação:
 - Entrada, saída, busca, ordenação, etc.

Exemplo: entrada e saída (1/5)

```
int v[10]; // Vetor de capacidade 10
int i;     // Variável para percorrer os índices
```

Exemplo: entrada e saída (2/5)

```
int v[10]; // Vetor de capacidade 10
int i;     // Variável para percorrer os índices

for(i = 0; i < 10; i++) {

}
```

Exemplo: entrada e saída (3/5)

```
int v[10]; // Vetor de capacidade 10
int i;     // Variável para percorrer os índices

for(i = 0; i < 10; i++){
    printf("Digite o %dº valor: ", i + 1 );
}
```

Exemplo: entrada e saída (4/5)

```
int v[10]; // Vetor de capacidade 10
int i;     // Variável para percorrer os índices

for(i = 0; i < 10; i++){
    printf("Digite o %dº valor: ", i + 1 );
    scanf("%d", &v[i]); // Entrada
}
```

Exemplo: entrada e saída (5/5)

```
int v[10]; // Vetor de capacidade 10
int i;     // Variável para percorrer os índices

for(i = 0; i < 10; i++){
    printf("Digite o %dº valor: ", i + 1 );
    scanf("%d", &v[i]); // Entrada
}

for(i = 0; i < 10; i++){
    printf("%d : %d\n", i, v[i]); // Saída
}
```

Considerações

- O exemplo é muito simples;
- Porém, mostra a rotina básica de entrada e saída para qualquer vetor:
 - com as devidas adaptações de *tipos, formatos e capacidades*.

Exemplo prático: limite do vetor

- Troque a condição de parada do 2º *for* (**i < 10**) por **i < 1000**;
- Analise a execução do programa.

Exemplo prático: limite do vetor

- Troque a condição de parada do 2º *for* (**$i < 10$**) por **$i < 1000$** ;
- Analise a execução do programa.
 - É esperado que alguns valores além do vetor sejam acessados na memória.
 - Porém, o programa trava em algum momento (acesso indevido à memória).

Atribuição de valores na declaração (1)

- Vetores podem ser inicializados na sua declaração;

Atribuição de valores na declaração (1)

- Vetores podem ser inicializados na sua declaração;
- Exemplo: vetor de capacidade 5 já inicializado:

```
float v[5] = {3.5, 7, 9, -1, 16};
```

Atribuição de valores na declaração (1)

- Vetores podem ser inicializados na sua declaração;
- Exemplo: vetor de capacidade 5 já inicializado:

```
float v[5] = {3.5, 7, 9, -1, 16};
```

<i>valores</i>	3.5	7	9	-1	16
<i>índices</i>	0	1	2	3	4

Atribuição de valores na declaração (2)

- É possível omitir a capacidade, caso o vetor seja inicializado;

Atribuição de valores na declaração (2)

- É possível omitir a capacidade, caso o vetor seja inicializado;
- Neste caso, a capacidade é definida implicitamente pelo compilador;

Atribuição de valores na declaração (2)

- É possível omitir a capacidade, caso o vetor seja inicializado;
- Neste caso, a capacidade é definida implicitamente pelo compilador;
- Exemplo: vetor de capacidade 8 definida implicitamente:

```
int vet[] = {2, 4, 6, 8, 10, 12, 14, 16};
```

Atribuição de valores na declaração (3)

- Vetores de caracteres (*strings*) podem ser inicializados de duas formas:

```
char nome1[] = { 'M', 'a', 'r', 'i', 'a' };
```

```
char nome2[] = "Maria";
```


Mais exemplos práticos

- Determinar o menor valor de um vetor;

Mais exemplos práticos

- Determinar o menor valor de um vetor;
- Determinar o menor valor de um vetor e o seu índice;

Mais exemplos práticos

- Determinar o menor valor de um vetor;
- Determinar o menor valor de um vetor e o seu índice;
- Ordenar os valores de um vetor usando o método da Seleção (*SelectionSort*).

Exercício em aula

- Faça um programa que lê 10 valores e os armazena em um vetor (*float*). Em seguida, mostre na tela:
 - O maior valor;
 - A média dos valores;
 - Quais valores estão acima da média.