INSTITUTO FEDERAL
BAIANO

Estrutura de Dados

Estruturas Homogêneas
Vetores eMatrizes

Prof. José Honorato Ferreira Nunes
honoratonunes@softwarelivre.org
http://softwarelivre.org/zenorato/honoratonunes

Estruturas Homogêneas – Matrizes Unidimensionais (Vetores)

Podemos definir um Vetor como uma variável dividida em vários "pedaços", em várias "casinhas", onde cada pedaço desses é identificado através de um número, referente à posição de uma determinada informação no vetor em questão.

O número de cada posição do vetor é chamado de índice.

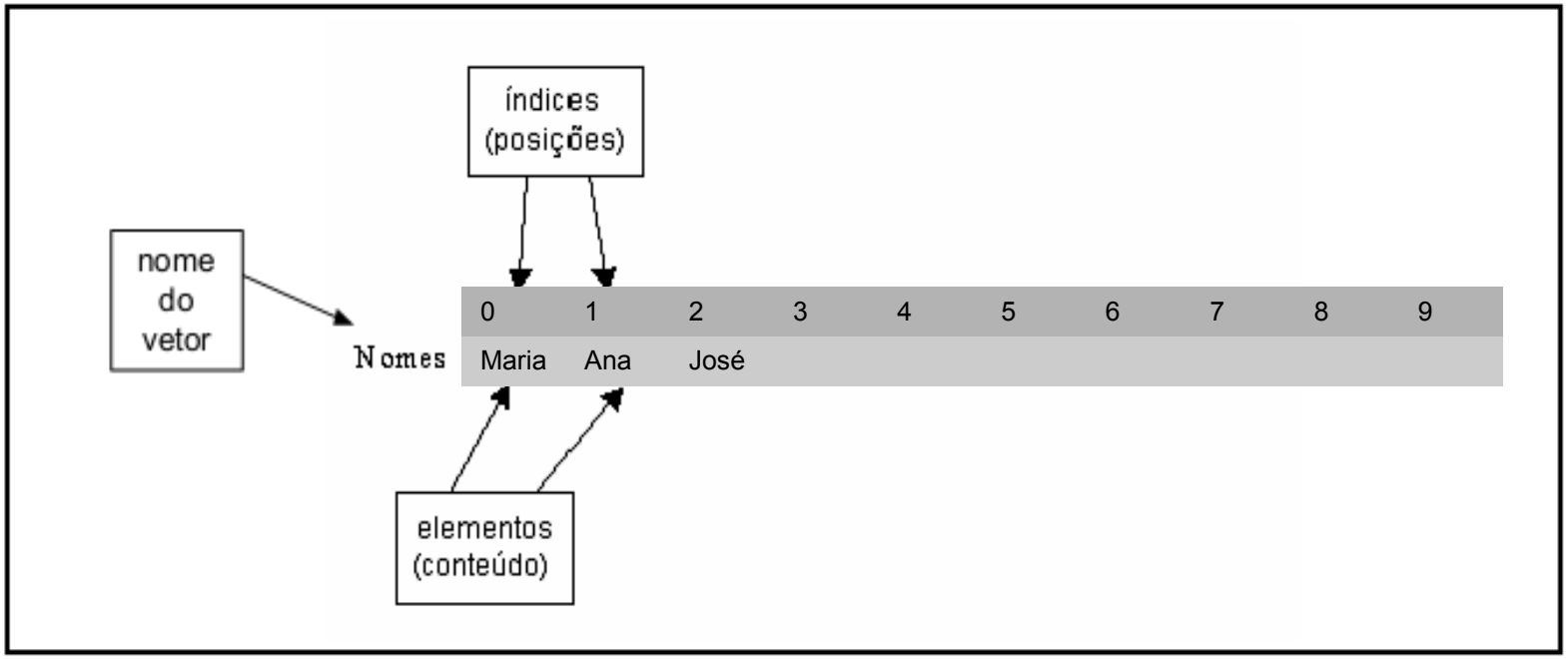
Estruturas Homogêneas – Matrizes Unidimensionais (Vetores)

Conceito: Vetor é um conjunto de variáveis, onde cada uma pode armazenar uma informação diferente, mas todas compartilham o mesmo nome.

São associados índices a esse nome, que representam as posições do vetor, permitindo assim, individualizar os elementos do conjunto.

Podemos imaginar que na memória do computador o vetor seja mais ou menos da seguinte forma:

Estruturas Homogêneas – Matrizes Unidimensionais (Vetores)



Vetores - declaração

Todos os elementos de um vetor pertencem necessariamente ao mesmo tipo de dado. Esta é a essência das estruturas de dados homogêneas.

No C, um vetor é declarado com a seguinte estrutura:

```
int valores[10];
char locais[5];
```

Vetores – atribuição e leitura

Para acessar (atribuir ou ler) um determinado elemento dentro do vetor, é necessário informar sua posição, também chamado de índice, por meio da seguinte sintaxe:

{para atribuir}

<nomeVetor>[índice] = <valor>;

valores[0] = 10;

{para ler}

<nomeVariavel>=<nomeVetor>[índice];

nota = valores[0];

Vetores - exemplos

```
#include <stdio.h>

int main () {
    int x, valores[10];

    for (x=0;x<=9;x++){
        printf("Informe um valor para posição %d: ", x+1);
        scanf("%d", &valores[x]);
    }
    for (x=0;x<=9;x++){
        printf("\n O elemento da posição %d vale: %d", x+1, valores[x]);
    }
    return 0;
}
```

Atividades

- Crie um algoritmo que solicite ao usuário 10 valores inteiros, armazenando os dados em um vetor. Em seguida, o algoritmo deverá percorrer o vetor escrevendo na tela os valores armazenados nas posições ímpares.

- Escreva um algoritmo que solicite ao usuário 10 valores inteiros e em seguida escreva na tela os valores lidos em ordem inversa.

Atividades

- Escreva um algoritmo para ler a nota de 30 alunos, calcular a média geral da turma e escrever quantos alunos tiveram a nota acima da média calculada.

- Faça um algoritmo que leia um vetor V[6]. Conte a seguir, quantos valores de V são negativos e mostre essa informação.

- Faça um algoritmo que leia um vetor C[15]. Encontre a seguir o maior elemento de C e mostre-o.

Matrizes

Matrizes são estruturas de dados que seguem o mesmo princípio dos vetores, porém as matrizes possuem duas ou mais dimensões, ao contrário dos vetores que possuem apenas uma dimensão.

A matriz mais comum é a de duas dimensões (linha e coluna), por se relacionar diretamente com a utilização de tabelas. Trabalharemos somente com matrizes de 2 dimensões, por serem mais comuns, mas podem ser necessárias, em algum momento, matrizes de 3 ou mais dimensões.

Matrizes

Uma matriz de 2 dimensões estará sempre fazendo menção a linhas e colunas e será representada por seu nome e tamanho.

Matriz Tabela		Coluna ↓						
		0	1	2	3	4	5	6
Linha →	0							
	1							
	2							
	3							
	4							

Matrizes - declaração

Todos os elementos de uma matriz pertencem necessariamente ao mesmo tipo de dado. Esta é a essência das estruturas de dados homogêneas.

No C, uma matriz é declarada com a seguinte estrutura:

```
int mat[5][7];
float num[2][4];
```

Matrizes – Atribuição e Leitura

Para acessar (atribuir ou ler) um determinado elemento dentro da Matriz, é necessário informar sua posição, também chamado de índice, por meio da seguinte sintaxe:

{para atribuir}

mat[0][0] = 10;

{para ler}

nota =mat[0][0];

Matriz - exemplo

```
#include <stdio.h>
int main () {
    int l, c, valores[10][5];
    for (l=0;l<=9;l++){
          printf("**** Preenchendo a linha %d : **** \n", l+1);
          for (c=0;c<=4;c++){
               printf("Informe um valor para coluna %d: ", c+1);
               scanf("%d", &valores[l][c]);
          }
    }
    for (l=0;l<=9;l++){
      for (c=0;c<=4;c++){
        printf("\n A posição [%d][%d] vale: %d", l+1,c+1, valores[l][c]);
      }
    }
    return 0;
}
```

Atividades

- Faça um algoritmo que leia uma matriz M[6,6] e uma matriz N[6,6]. A seguir, calcule o produto de M por N, colocando os resultados em uma matriz PROD[6,6].

- Faça um algoritmo que leia uma matriz S[5,5] e um valor A. A seguir, multiplique a matriz S pelo valor A, colocando o resultado em um vetor V[25]. Mostre a matriz S[5,5] e o vetor V[25].

Atividades

- Leia uma matriz M[5,5] e crie 2 vetores SL[5] e SC[5] que contenham respectivamente as somas das linhas e das colunas de M.
- Leia uma matriz M[6,5]. Após, divida os 5 elementos de cada linha da matriz pelo maior elemento de cada uma das 6 linhas. Coloque o resultado em uma matriz S[6,5].
- Escreva um algoritmo que lê uma matriz M[6,6]. A seguir, troque os elementos da primeira coluna com os elementos da segunda coluna, os da terceira coluna com a quarta coluna e os elementos da quinta coluna com os elementos da sexta coluna.

Bibliografia Básica

- MANZANO,Wilson Y. Yamaturni - São Paulo-SP. **Lógica estruturada para programação de computadores**, Ed. Érica 1997 e 2001.

- MORAES, Celso Roberto. **Estruturas de Dados e Algoritmos**. Ed. Érica, São Paulo

- LOPES, Anita. **Introdução à programação**. Rio de Janeiro: Campus,2002.

Bibliografia Complementar

- BENEDUZZI, Humberto M. e METZ, João A. **Lógica e Linguagem de Programação – Introdução ao Desenvolvimento de Software** (1ª edição). Editora do Livro Técnico, 2010

- FORBELLONE, A. L. V. e Eberspacher, H. F**. Lógica de Programação - a Construção de Algoritmos e Estruturas de Dados** (3ª edição). Pearson, 2005

- CORMEN, Thomas H. et. al. **Algoritmos: Teoria e Prática**. Editora Campus, 2002.

- ZIVIANI, Nivio. **Projeto de Algoritmos**. Editora Nova Fronteira, 2004.

- SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. Bookman, 2001.

Prof. José Honorato F. Nunes
honoratonunes@gmail.com