Algoritmos e Lógica de Programação

- Douglas Baptista de Godoy

Fatec
Faculdade de Tecnologia

Etec
Escola Técnica Estadual

Centro Paula Souza

GOVERNO DO ESTADO
DE SÃO PAULO

Ementa

- Projeto e representação de algoritmos.
- Estruturas de controle de fluxo de execução: sequência, seleção e repetição.
- Tipos de dados básicos e estruturados (vetores e registros).
- Rotinas. Arquivos.
- Implementação de algoritmos usando uma linguagem de programação.

Objetivo

- Analisar problemas computacionais e projetar soluções por meio da construção de algoritmos.

Avaliação

- Nota1 - Avaliar os conhecimentos adquiridos no 1º bimestre - Nota 1
  - 08/04/2021
- Nota2 - Avaliar os conhecimentos adquiridos no 2º bimestre - Nota 2
  - 17/06/2021
- Recuperação - Substituíra a menor nota do aluno. O aluno só poderá fazer se tirar menos do que 6 em uma das duas provas. - Recuperação abordando todo o conteúdo da disciplina no semestre.
  - 24/06/2021

Vetores e Matrizes

- **Definição de Vetor**
- As variáveis compostas homogêneas unidimensionais (ou, simplesmente, vetores) são capazes de armazenar diversos valores. Cada um desses valores é identificado pelo mesmo nome (o nome dado ao vetor), sendo diferenciado entre si apenas por índice.
- Os índices utilizados na linguagem C/C++ para identificar as posições de um vetor começam sempre em 0 (zero) e vão até o tamanho do vetor menos uma unidade. O índice de um vetor em C/C++ deve sempre ser representado por um dos tipos inteiros disponíveis na linguagem.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

Vetores e Matrizes

- **Declaração de Vetor**
- Os vetores em C/C++ são identificados pela existência pela existência de colchetes logo após o nome da variável no momento da declaração. Dentro dos colchetes, deve-se colocar o numero de elementos que o vetor poderá armazenar.
- Em C/C++, a indicação do tamanho do vetor (ou seja, a quantidade de elementos que o vetor poderá armazenar) deve ser feita por um inteiro fixo(representado por um literal ou uma constante). Se houver necessidade de definir o tamanho do vetor em tempo de execução, deve-se fazê-lo através de ponteiros.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

9. Vetores e Matrizes

- **Exemplo de Vetor**

Declaração de Vetor

Exemplo 1:
int vet[10];

Exemplo 2:
char x[5];

Exemplo 3:
#define tam 5;
char z[tam];

Atribuindo valores ao Vetor

Exemplo 1:
vet[0] = 1;

Exemplo 2:
x[3] = 'b' ;

Exemplo 3:
for (i = 0; i < 10; i++)
*scanf("%d%*c", &vet[i]);*

Mostrando elementos do Vetor

Exemplo 1:
printf("%d", vet[5]);

Exemplo 2:
printf("%c", x[3]);

Exemplo 3:
for (i = 0; i < 10; i++)
printf("%d", vet[i]);

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

Vetores e Matrizes

- **Definição de Matriz**
- Uma Matriz pode ser definida como um conjunto de variáveis de mesmo tipo e identificadas pelo mesmo nome. Essas variáveis são diferenciadas por meio da especificação de suas posições dentro dessa estrutura.
- A linguagem C/C++ permite a declaração de matrizes unidimensionais, bidimensionais e multidimensionais. O padrão ANSI prevê ate 12 dimensões. Entretanto, o limite de dimensões fica por conta da quantidade de recursos computacionais disponíveis. Apesar disso, as matrizes mais utilizadas possuem duas dimensões. Para cada dimensão deve ser utilizado um índice.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

Vetores e Matrizes

- **Definição de Matriz**

- Os índices usados na linguagem C/C++, para identificar as posições de uma matriz, começam sempre em 0 (zero) e vão até o tamanho da dimensão menos uma unidade. Os índices de uma matriz em C/C++ devem sempre ser representados por um dos tipos inteiros disponíveis na linguagem.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

Vetores e Matrizes

- **Declaração de Matriz**
- *tipo_dos_dados nome_variável [dimensão1][dimensão2][....][dimensãoN];*
- Onde:
- tipo_dos_dados: é o tipo dos dados que serão armazenados na matriz;
- nome_variável: é o nome dado à variável do tipo matriz;
- [dimensão1]: representa o tamanho da 1ª dimensão da matriz;
- [dimensão2]: representa o tamanho da 2ª dimensão da matriz;
- [dimensãoN]: representa o tamanho da n-ésima dimensão da matriz;

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

Vetores e Matrizes

- **Declaração de Matriz**

- Em C/C++, a indicação do tamanho das dimensões de uma matriz deve ser feita por um valor inteiro fixo (representado por um literal ou uma constante). Se houver necessidade de definir o tamanho da matriz em tempo de execução, deve-se faze-lo por meio de ponteiros.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

Vetores e Matrizes

- **Exemplo de Matriz**

Declaração de Matriz

Exemplo 1:
float X[2][6];

Exemplo 2:
char MAT[4][3];

Exemplo 3:
float Y[2][4][3];

Atribuindo valores a Matriz

Exemplo 1:
X[1][4] = 5;

Exemplo 2:
MAT[3] [2]= 'D' ;

Exemplo 3:
Y[0][3][1] = 12;

Exemplo 4:

```
for (i = 0; i < 2; i++)
{
  for (j = 0; j < 6; j++)
       scanf("%f%*c", &X[i][j]);
}
```

Mostrando elementos da Matriz

Exemplo 1:
printf("%f", X[1][4]);

Exemplo 2:
printf("%c", MAT[3][2]);

Exemplo 3:

```
for (i = 0; i < 3; i++)
{
  for (j = 0; j < 6; j++)
       printf("%f", X[i][j]);
}
```

Referencias Bibliográficas

- ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de, **Fundamentos da Programação de Computadores**, Pearson Editora, 3ª edição.