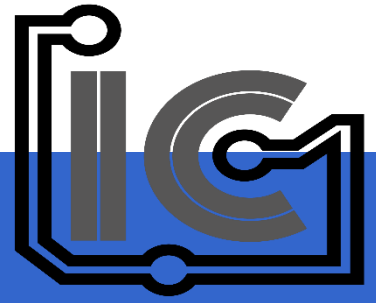
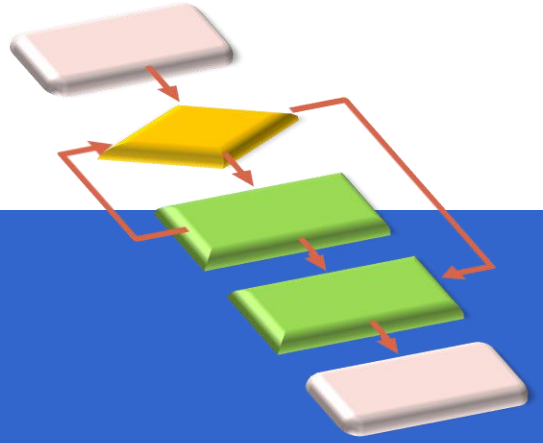


INSTITUTO DE COMPUTAÇÃO



Algoritmos II

CONCEITOS INTRODUTÓRIOS
Dados estruturados

Prof.ª Vanessa de Oliveira Campos

Dados estruturados

- Arranjos unidimensionais;
- Arranjos multidimensionais;
- Registros.

Arranjo unidimensional: Vetor

- **Vetor** é uma variável composta homogênea **unidimensional** formada por uma *sequência de variáveis*, todas do *mesmo tipo*, com o mesmo identificador e alocado sequencialmente na memória.

- Na linguagem C, a declaração de um vetor segue a sintaxe:

<tipo> identificador [<tamanho do vetor>];

Vetor

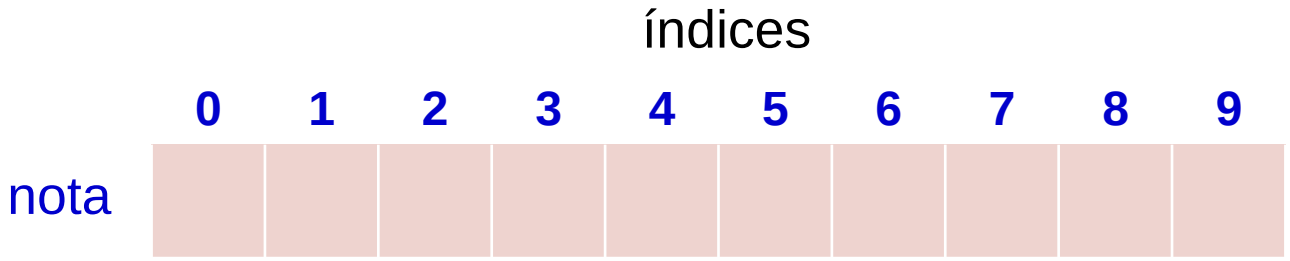
- Exemplo:

```
float nota[10];
int medias [40];
```

- Os índices utilizados na linguagem C para identificar as posições de um vetor começam sempre em 0 (zero) e vão até o tamanho do vetor menos uma unidade.
- Acesso por meio de um índice inteiro.
- Posições contíguas na memória.
- Tamanho pré-definido.
- Índices fora dos limites podem causar comportamento anômalo do programa.

Vetor

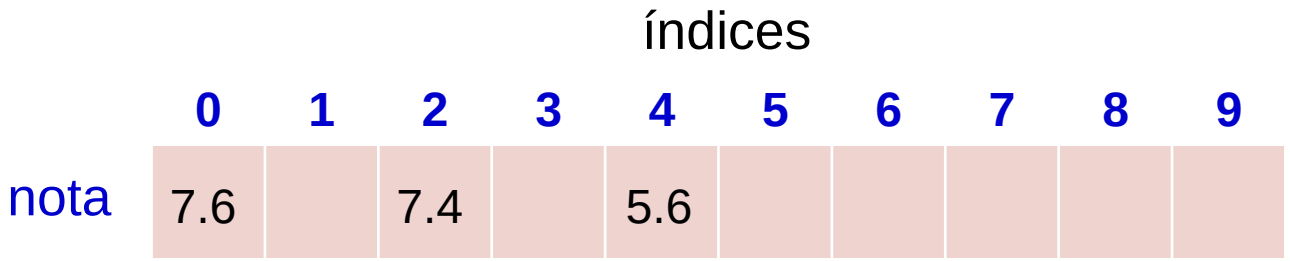
```
float nota[10];
```



Características de um vetor

Vetor

```
float nota[10];
nota[2] = 7.4;
nota[4] = 5.6;
nota[0] = nota[4] + 2;
```



Vetor

- Preencher um vetor significa atribuir valores a todas as suas posições.
- Assim, deve-se implementar um mecanismo que controle o valor do índice.

```
float nota[10];
int indice;
for (indice = 0; indice < 10 ; indice++ )
{
   scanf("%f", &nota[indice]);
}
```

- O acesso de um vetor em uma posição específica tem o mesmo comportamento que uma variável simples.

Arranjo multidimensional: Matriz

- É uma variável composta homogênea **multidimensional** formada por uma sequência de variáveis, todas do ***mesmo tipo***, com o mesmo identificador e alocadas sequencialmente na memória.
- A distinção entre as variáveis é feita usando *índices* para **cada dimensão** da matriz.

Arranjo multidimensional: Matriz

- Na linguagem C, a declaração de uma matriz segue a sintaxe:

<tipo> identificador [<linhas>] [<colunas>];

- Uma matriz possui *linhas* x *colunas* variáveis do tipo <tipo>.
- As linhas são numeradas de 0 a *linhas* − 1.
- As colunas são numeradas de 0 a *colunas* − 1.
- Como exemplo, em uma Turma, a declaração poderia ser:

```
float notasturma[19][4];
```

Matriz: acesso aos elementos

- O acesso a um elemento da matriz é realizado da seguinte forma:

  nome_da_matriz [<linha>] [<coluna>]

- No nosso exemplo, para acessar a sua 4ª nota (coluna) do 2º aluno (linha):

```
float nota = notasturma[1][3];
```

```
int i, j;
i = 1;
j = 3;
float nota = notasturma [i][j];
if (nota = notasturma [i+1][j])
   printf("notas iguais\n");
```

Registro

- **Registros** são *estruturas heterogêneas*, ou seja, *coleções de dados* de quaisquer tipos.
  - Os dados que integram um registro são denominados campos.
  - Os registros e seus campos devem ser designados com nomes que sejam únicos.

Registro

- Sintaxe:

```
struct < nome do registro > {
        <tipo do campo1>  <nome do campo1> ;
        <tipo do campo2>  <nome do campo2> ;
        …
        <tipo do campoN>  <nome do campoN> ;
};
```

Registro

- Exemplo:

```
struct funcionario {
    int cod;
    char nome[30];
    float salario;
    int depto;
    char cargo;
};
```

Registro

```
struct data {
    int dia;
    char mes[3];
    int ano;
};

struct pessoa {
    char nome[10];
    struct data dia_admissao;
    float salario;
};

struct data admissao;
struct pessoa funcionario_com_data;
```

Registro

- Sintaxe:

```
struct {
        <tipo do campo1>  <nome do campo1> ;
        <tipo do campo2>  <nome do campo2> ;
        …
        <tipo do campoN>  <nome do campoN> ;
} <nomes de variáveis separados por vírgulas>;
```

Registro

- Exemplo:

```
struct {
    int cod;
    char nome[30];
    float salario;
    int depto;
    char cargo;
} um_funcionario, outro_funcionario;
```

Registro

- Referência aos campos de uma estrutura

  <nome da variável>.<nome do campo>

- Exemplo:

```
/* preencher campo nome por leitura: */
scanf("%s", &um_funcionario.nome);
/* atribuir valor ao seg. elemento do vetor do campo salario: */
um_funcionario.salario[1] = 5800.00;
/* comparar valor de um campo */
if (funcionario.cargo == 'A')
   funcionario.depto = 10;
```