



# Computação 2

## Aula 2

### Vetores e Matrizes

Prof<sup>a</sup>. Fabiany  
fabiany1@utfpr.edu.br

# Vetores

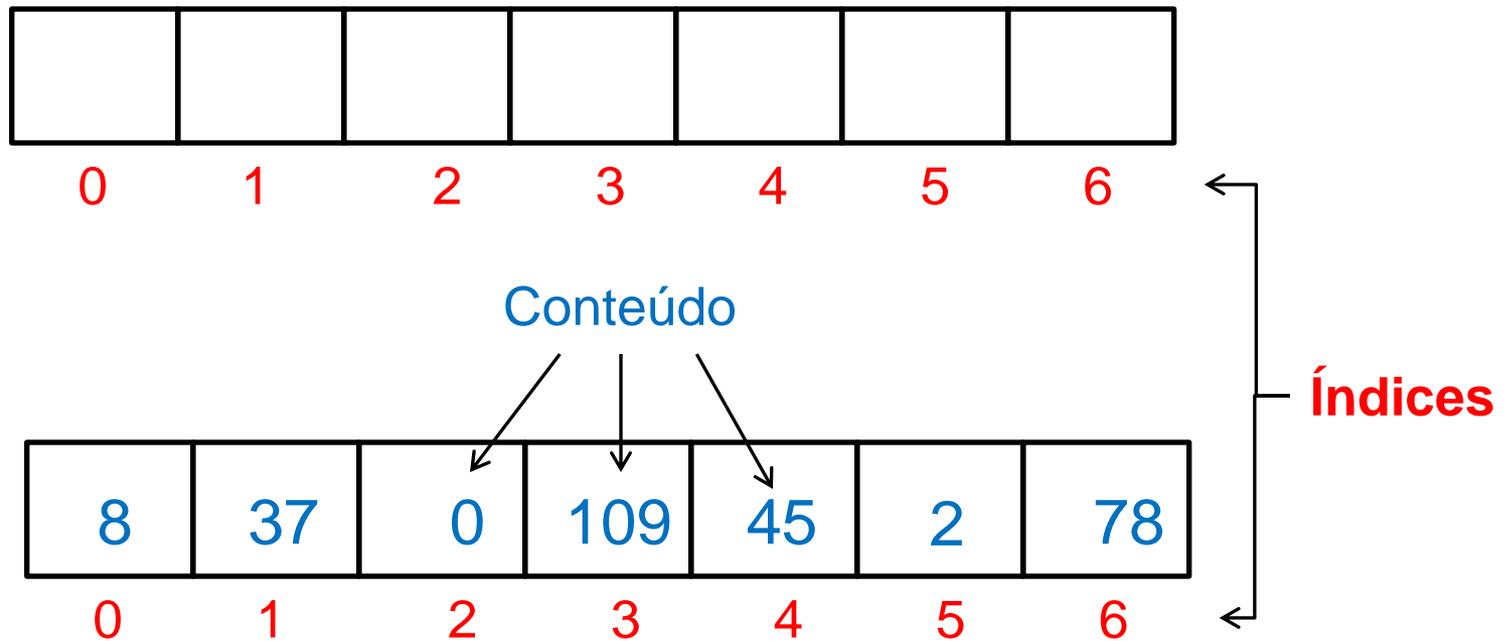
- Vetor também é conhecido como variável composta homogênea unidimensional;
- Um vetor computacional é um “variável composta” capaz de armazenar uma quantia determinada de valores;
- Um vetor computacional é inspirada no vetor matemático, que também é capaz de armazenar um conjunto de valores.

# Vetores

- Um conjunto de variáveis do mesmo tipo, que possuem o mesmo identificador(nome) e são alocados seqüencialmente na memória;
- Como as variáveis tem o mesmo nome, o que as distingue é um índice que referencia sua localização dentro da estrutura.

# Vetores

Exemplo de vetor:



# Vetores

- Declaração de um vetor:
- `<tipo> NOME_VAR [tamanho];`

Onde :

**tamanho** representa a capacidade de armazenamento do vetor;

**< tipo >** - representa qualquer um dos tipos básicos ou tipo anteriormente definido (int, float, double, char).

# Vetores

- Exemplo de declaração:

**float** notas[4];

**float** valor[15], x[2];

**int** idade[50], valores[10];

**char** letras[50];

# Vetores

- Exemplos de declaração e inicialização:

```
float notas[4] = {7.0,8.5,9,9.5}
```

```
int valores[ ] = {1,2,3,4,5,6,7,8,9,10}
```

```
float vetorx[45];
```

```
int idade[3];
```

# Vetores

- Atribuindo valores ao vetor:
  - As atribuições em vetor exigem que seja informada em qual de suas posições o valor ficará armazenado.

notas[1] = 56;

notas[5] = 100;

x[0] = 8.9;

x[1] = 0.5;

letras[3] = 'a'; letras[20] = '\*';

# Vetores

```
void main()
```

```
{
```

```
    float vet_nums [5];
```

```
    // 5 posições, de 0 à 4... (sempre começa por zero em C).
```

```
    float media, quant;
```

```
    vet_nums[0] = 9.0;
```

```
    vet_nums[1] = 8.0;
```

```
    vet_nums[2] = 7.0;
```

```
    vet_nums[3] = 7.0;
```

```
    vet_nums[4] = 7.0;
```

```
    quant = vet_nums[0] + vet_nums[1] + vet_nums[2] + vet_nums[3] + vet_nums[4];
```

```
    media = quant / 5.0;
```

```
    printf (“ A média dos 5 números é: %f.”, media);
```

```
}
```

```
void main ( )
{
    float vet_nums[5];
    float media, quant;

    printf ( “Digite o 1º número: \n” );
    scanf ( “%f”, &vet_nums[0] );

    printf ( “Digite o 2º número: \n” );
    scanf ( “%f”, &vet_nums[1] );

    printf ( “Digite o 3º número: \n” );
    scanf ( “%f”, &vet_nums[2] );

    printf ( “Digite o 4º número: \n” );
    scanf ( “%f”, &vet_nums[3] );

    printf ( “Digite o 5º número: \n” );
    scanf ( “%f”, &vet_nums[4] );

    quant = vet_nums[0] + vet_nums[1] + vet_nums[2] + vet_nums[3] + vet_nums[4];
    media = quant / 5.0;

    printf ( “ A média dos 5 números é: %f.”, media);
}
```

# Exemplo Vetores

```
void main ()
{

    float vet_nums[5];
    float media, quant;
    int  cont;

    quant = 0;

    for ( cont = 0; cont < 5; cont = cont + 1 )
    {
        printf ( "Digite o %i .o número:", cont+1 );
        scanf ( "%f", &vet_nums[ cont ] );

        quant = quant + vet_nums [ cont ] ;
    }

    media = quant / 5.0;

    printf ( " A média dos 5 números é: %f.", media);

}
```

# Vetores

- **Preenchendo um vetor:**

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

```
void main()
{
    int i, vet[5];
    for(i=0;i<5;i=i+1){
        printf("Digite um valor\n");
        scanf("%i", &vet[i]);
    }
}
```

# Vetores

- Mostrar um elemento de um vetor, também é necessário utilizar índices:

```
void main()
{
    int i;
    int valores[5] = {1,2,3,4,5};
    for(i=0;i<5;i=i+1){
        printf("O valor da posicao %i é %i \n", i, valores[i]);
    }
}
```

# Exercícios

- 1) Faça um programa para ler 15 idades diferentes e guarde em um vetor. Depois calcular e mostrar a média das idades.
- 2) Faça um programa que entre com 10 números pelo teclado; armazene-os em um vetor de dez posições; imprima-o na tela na ordem em que os números foram entrados e depois na ordem inversa.
- 3) Faça um programa que preencha dois vetores de dez elementos inteiros cada e mostre um terceiro vetor resultante da intercalação deles.
- 4) Faça um programa para ler 10 números e guardar em um vetor. Depois calcular e mostrar qual número é par e qual seu índice no vetor..

# Exercícios

5) Faça um programa para ler a idade de 50, classifique-o de acordo com as seguintes categorias:

- infantil A = 5 – 7 anos;
- infantil B = 8 – 10 anos;
- juvenil A = 11 – 13 anos;
- juvenil B = 14 – 17 anos;
- adulto = maiores ou igual a 18 anos.

A seguir calcule e mostre a média de idade dos nadadores, a idade do mais velho e a idade do mais jovem. Mostrar quantos nadadores têm idade maior que a média.

# Matrizes

- Matriz também é conhecido como variável composta homogênea multidimensional;
- Uma matriz computacional é uma “variável composta” capaz de armazenar uma quantidade determinada de valores;
- Uma matriz computacional é inspirada na matriz matemático, que também é capaz de armazenar um conjunto de valores.

# Matrizes

- Um conjunto de variáveis do mesmo tipo, que possuem o mesmo identificador(nome) e são alocados seqüencialmente na memória;
- Como as variáveis tem o mesmo nome, o que as distingue é um índice que referencia sua localização dentro da estrutura.
- Uma variável do tipo matriz precisa de um índice para cada uma de suas dimensões.

# Matrizes

Exemplo de Matriz:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 0 |   |   |   |   |   |   |   |
| 1 |   |   |   |   |   |   |   |
| 2 |   |   |   |   |   |   |   |
|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

# Matrizes

- Declaração de uma matriz:
- `<tipo> NOME_MATRIZ [Tamanho1] [Tamanho2];`
- Onde:
- *Tamanho1* e *Tamanho2* representam as dimensões da matriz

`< tipo >` - representa qualquer um dos tipos básicos ou tipo anteriormente definido.

# Exemplo Matriz

```
#include <stdio.h>
```

```
void main ( )
```

```
{
```

```
float matriz [2] [2];
```

```
float media, quant;
```

```
matriz [0][0] = 9.0;
```

```
matriz [0][1] = 8.0;
```

```
matriz [1][0] = 7.0;
```

```
matriz [1][1] = 7.0;
```

```
quant = matriz[0][0] + matriz[0][1] + matriz[1][0] +  
matriz[1][1];
```

```
media = quant / 4.0;
```

```
printf ( " A média dos 4 números é: %f.", media );
```

```
}
```



| Índices | 0          | 1          |
|---------|------------|------------|
| 0       | <b>9.0</b> | <b>8.0</b> |
| 1       | <b>7.0</b> | <b>7.0</b> |

```
#include <stdio.h>
void main ( )
{
    float matriz [2][2];

    float media, quant;

    printf ( “ Digite o 1o número: ” );
    scanf ( “%f”, &matriz[0][0] );

    printf ( “ Digite o 2o número: ” );
    scanf ( “%f”, &matriz[0][1] );

    printf ( “ Digite o 3o número: ” );
    scanf ( “%f”, &matriz[1][0] );

    printf ( “ Digite o 4o número: ” );
    scanf ( “%f”, &matriz[1][1] );

    quant = matriz [0][0] + matriz [0][1] + matriz [1][0] + matriz [1][1];

    media = quant / 4.0;
    printf ( “ A média dos 4 números é: %f.”, media );
}
```

# Declaração de Vetores e Matrizes

Exemplos:

```
int main()  
{
```

```
    float parc_aluno[2];
```

```
    float notas_alunos[5][2];
```

```
    char nome[40];
```

```
    char nomes_turma[5][40];
```

```
    return 0;
```

```
}
```

# Exemplo

- Elaborar um programa que leia as duas parciais de cada aluno de uma turma de 30 alunos, classificando cada um como aprovado, em exame ou reprovado.
  - O aluno que obtiver média maior ou igual a 7 será aprovado.
  - O aluno que obtiver 10 na segunda parcial também será aprovado.
  - O aluno que obtiver média maior ou igual a 5 irá para exame.
  - O aluno que obtiver média maior ou igual a  $P$  também irá para exame, sendo  $P$  calculado como  $1,5 \times \text{média\_da\_turma}$ .

# Solução

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int cont;
    float parc_aluno[2];

    float med_aluno = 0.0;
    float med_turma = 0.0;
    float soma      = 0.0;
    float parametro = 0.0;

    float notas_alunos[5][2];

    for (cont = 0; cont < 5; cont++)
    {
        printf ("Primeira nota: \n");
        scanf ("%f", &parc_aluno[0]);
        printf ("Segunda nota: \n ");
        scanf ("%f", &parc_aluno[1]);

        notas_alunos[cont][0] = parc_aluno[0];
        notas_alunos[cont][1] = parc_aluno[1];
        med_aluno = ( parc_aluno[0] + parc_aluno[1] )/2.0;
        soma = soma + med_aluno;
        printf("\n");
    }

    med_turma = soma / 5.0;
    parametro = (1.5)*med_turma;
    printf ("A média da turma é: %f \n", med_turma);
    printf ("O parametro é : %f \n \n", parametro);
}
```

```
for (cont = 0; cont < 5; cont++)
{
    med_aluno = ( notas_alunos [cont][0] +
                 notas_alunos [cont][1] )/2.0;

    if ( (med_aluno >= 7) || (10 == notas_alunos[cont][1]) )
    {

        printf ("Aluno %d aprovado! \n", cont);

    }
    else
    {
        if ( (med_aluno >= 5) || (med_aluno >= parametro) )
        {

            printf ("Aluno %d em exame! \n", cont );

        }
        else
        {

            printf ("Aluno %d reprovado! \n", cont);

        }
    }
}

system ("pause");
return 0;
}
```

# Exercícios

- Faça um programa que preencha uma matriz  $10 \times 10$  com números inteiros e depois multiplique cada elemento da matriz por 2 e guarde em uma nova matriz e depois mostre o resultado.
- Faça um programa para ler 4 notas de 3 alunos e guarde os valores em uma matriz. Depois calcular e mostrar a média das notas de cada aluno.

# Exercícios

- Faça um programa que preencha duas matrizes  $8 \times 8$  com números inteiros e depois multiplique a diagonal principal da primeira pela diagonal secundária da segunda. Depois mostre o resultado.