

Linguagem C: vetores/matrizes

Prof. Críston
Algoritmos e Programação

Vetores/matrizes

- Recurso das linguagens que facilita a declaração/acesso/manipulação de uma sequência de variáveis de um mesmo tipo
- Um vetor é uma matriz com apenas 1 dimensão
- Em alguns livros esta estrutura é chamada de **homogênea**, pois armazena elementos de um mesmo tipo
 - Quando a estrutura permite elementos de tipos diferentes, é chamada de **heterogênea** (veremos posteriormente)

Vetores - declaração

```
tipo nome [ tamanho ];
```

- Exemplos:

```
int idades[30];
```

```
double alturas[30];
```

Vetores - acessando elementos

- Os elementos são acessados com o operador [], onde o primeiro elemento está na posição 0, o segundo na posição 1, assim sucessivamente até a posição tamanho-1
- Exemplos:

```
altura[2] = 1.5;
if (idade[20] > 18) ...
for (i=0; i<30; i++) printf("%f\n", altura[i]);
```
- **A linguagem C não verifica se você está utilizando índice inválido! Você pode acabar lendo lixo ou alterando outras variáveis!**

Exemplo

- Leia 10 números inteiros positivos e armazene em um vetor. Em seguida determine e imprima o maior elemento no vetor.

```
main()
{
    int v[10], i, maior = 0;
    for (i=0; i<10; i++)
        scanf("%d", &v[i]);
    for (i=0; i<10; i++)
        if (v[i] > maior)
            maior = v[i];
    printf("maior = %d\n", maior);
}
```

Vetores – inicialização na declaração

```
int vetor[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

- Esta inicialização pode ser feita apenas na declaração
- Se o num de elementos é menor que o tamanho do vetor, os elementos restante são inicializados com zero
- Podemos omitir o tamanho do vetor quando ele é inicializado

```
int vetor[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

- Sem inicialização o tamanho é obrigatório

```
int vetor[];
```

Vetores multidimensionais

- Declaração

```
tipo nome [tam1][tam2]...[tamN];
```

- Inicialização

```
float matriz [3][4] = {1,2,3,4,5,6,7,8,9,10,11,12};
```

```
float matriz [ ][4] = {1,2,3,4,5,6,7,8,9,10,11,12};
```

- Apenas a primeira dimensão pode ser omitida

```
float matriz [ ][ ] = {1,2,3,4,5,6,7,8,9,10,11,12};
```

Exemplo

- Leia os elementos de uma matriz 3x4 de números float. Em seguida imprima os elementos no formato abaixo:

```
X.XX      X.XX      . . .      X.XX
X.XX      X.XX      . . .      X.XX
. . .      . . .      . . .      . . .
X.XX      X.XX      . . .      X.XX
```

10 colunas por elemento.

Exemplo

```
main()
{
    float m[3][4];
    int i, j;
    for (i=0; i<3; i++)
        for (j=0; j<4; j++)
            scanf("%f", &m[i][j]);
    for (i=0; i<3; i++)
    {
        for (j=0; j<4; j++)
            printf("%10.2f", m[i][j]);
        printf("\n");
    }
}
```

Exercício

- Leia duas matrizes 2×3 de números double. Imprima a soma destas duas matrizes.
- Leia uma matriz 3×2 e outra 2×3 de números double. Imprima o produto destas duas matrizes.
- Conte os votos de uma eleição com 10 candidatos. Leia os votos de um arquivo. Use um vetor para os contadores.
- Uma cidade tem 4 ruas leste-oeste e 6 ruas norte-sul. Ou seja, tem $4 \times 6 = 24$ cruzamentos. Leia uma seqüência de pares de ruas que indicam o cruzamento onde cada acidente ocorreu. Imprima uma matriz com a contagem de acidentes de cada cruzamento.