Aula 9: Matrizes

Adriano Veloso

Algoritmos e Estruturas de Dados I - DCC/UFMG

Matrizes

- Arranjos podem ter mais de um índice.

Matrizes

- Arranjos podem ter mais de um índice.
- Arranjos de 2 índices são chamados de matrizes.

Matrizes

- Arranjos podem ter mais de um índice.
- Arranjos de 2 índices são chamados de matrizes.
- Arranjos de 3 índices são chamados de cubóides.

Matrizes

- Arranjos podem ter mais de um índice.
- Arranjos de 2 índices são chamados de matrizes.
- Arranjos de 3 índices são chamados de cubóides.

Serve especialmente para agrupar dados de forma a facilitar o acesso rápido.

Declaração de matrizes

Uma matriz é uma variável

- Tem um nome, a partir do qual podemos acessar todos os seus elementos
- Cada elemento é acessível através de dois índices
- O índice é um inteiro, especificado entre colchetes []
- As dimensões da matriz têm um tamanho máximo

Declaração de matrizes

Uma matriz é uma variável

- Tem um nome, a partir do qual podemos acessar todos os seus elementos
- Cada elemento é acessível através de dois índices
- O índice é um inteiro, especificado entre colchetes []
- As dimensões da matriz têm um tamanho máximo

Quando uma matriz é declarada, precisamos especificar seu nome, tamanho e tipo.

Exemplo

Você pode armazenar valores de todos tipos, por exemplo inteiros ou reais.

```
float notas[3][55];
```

Exemplo

Você pode armazenar valores de todos tipos, por exemplo inteiros ou reais.

```
float notas[3][55];
```

Para percorrer essa matriz você pode usar estruturas de iteração:

```
for(int i=0;i<3;i++){
    for(int j=0;j<55;j++){
        printf("A nota do aluno %d na prova %d é %f", j, ,i, notas[i][j]);
    }
}
```

Matrizes na Memória

Quando você declara uma matriz, uma certa quantidade de memória é alocada.

- Basicamente, as dimensões da matriz multiplicado pelo número de bytes de seu tipo.
- A memória é alocada de forma contígua (posições sub-sequentes de memória). Por isso a rapidez de acesso.

Matrizes na Memória

Quando você declara uma matriz, uma certa quantidade de memória é alocada.

- Basicamente, as dimensões da matriz multiplicado pelo número de bytes de seu tipo.
- A memória é alocada de forma contígua (posições sub-sequentes de memória). Por isso a rapidez de acesso.
- A variável é um ponteiro que indica o começo dessa memória alocada.

Matrizes na Memória

Quando você declara uma matriz, uma certa quantidade de memória é alocada.

- Basicamente, as dimensões da matriz multiplicado pelo número de bytes de seu tipo.
- A memória é alocada de forma contígua (posições sub-sequentes de memória). Por isso a rapidez de acesso.
- A variável é um ponteiro que indica o começo dessa memória alocada.

Uma boa forma é se pensar que uma matriz é um arranjo de arranjos.

notas[0][0], notas[0][1], ..., notas[2][53], notas[2][54]

Cuidados

Se você declara:

int laranjas[2][3];

O que contece se você tenta acessar o conteúdo de laranjas[0][5]?

Cuidados

Se você declara:

int laranjas[2][3];

O que contece se você tenta acessar o conteúdo de laranjas[0][5]?

Você pode acessar um valor qualquer, ou até mesmo acessar uma posição de memória "proibida", produzindo erros de execução.

Funções e matrizes

- Também podemos passar uma matriz como parâmetro.

Funções e matrizes

- Também podemos passar uma matriz como parâmetro.
- Nesse caso o que é passado realmente é o endereço da matriz.

Funções e matrizes

- Também podemos passar uma matriz como parâmetro.
- Nesse caso o que é passado realmente é o endereço da matriz.

```
int soma_matriz(int vendas[][]);

int soma_matriz(int** vendas);
```

Exemplo

```
int main(){
    int vendas[2][3]={{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int s=soma_matriz(vendas);
    printf("A soma é: %d", s);
}
```

Exemplo

```
int main(){
    int vendas[2][3]={{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int s=soma_matriz(vendas);
    printf("A soma é: %d", s);
}
int soma_matriz(int mat[][]){
    int soma=0;
    for(int i=0;i<2;i++) {
        for(int j=0;j<3;j++) {
            soma+=mat[i][j];
        }
    }
    return(soma);
}
```

Exemplo

Você pode iterar por colunas e depois por linhas.

Exemplo

Você pode iterar por colunas e depois por linhas.

```
int soma_matriz(int mat[][]){
    int soma=0;
    for(int j=0;j<3;j++) {
        for(int i=0;i<2;i++) {
            soma+=mat[i][j];
        }
    }
    return(soma);
}
```

Contato

adrianov@dcc.ufmg.br