Programação

IFMG CODAAUT

Prof. Marco Antonio M. Carvalho

UFOP
Universidade Federal de Ouro Preto

PROEX
Pró-reitoria de extensão da UFOP

decom
departamento de computação

INSTITUTO FEDERAL
MINAS GERAIS

Lembretes

- **Lista de discussão**
  - Endereço:
    - programaacao@googlegroups.com
  - Solicitem acesso:
    - http://groups.google.com/group/programaacao
- **Página com material dos treinamentos**
  - http://www.decom.ufop.br/marco/extensao/obi/
- **Repositório online de problemas das edições passadas da OBI**
  - http://br.spoj.com/problems/obi/sort=-7
- **Moodle**
  - http://programaacao.net.br/login/index.php

Avisos

- Redirecionamento da entrada
  - Os problemas da OBI listados no SPOJ não exigem redirecionamento da entrada;
  - No entanto, outros problemas do SPOJ exigem;
  - Foi disponibilizado um material suplementar na página e no *moodle* do curso com instruções a este respeito.

Na aula de hoje

- EOF;
- Matrizes;
- Problemas Selecionados;
- Um Problema de Lógica.

EOF

EOF

- Uma forma simples de resolver este problema é simular o marcador de final de arquivo (**EOF**)
  - No windows, basta digitar ctrl+z após digitar os dados de entrada;
  - Em distribuições Unix, basta digitar ctrl+d após digitar os dados de entrada.
- Observe que a combinação de teclas pode variar de acordo com o sistema operacional.

Matrizes

Matrizes

- Vetores podem ser entendidos como matrizes de uma dimensão;
- E matrizes podem ser entendidas como vetores de **mais de uma** dimensão
  - É como se cada posição de um vetor fosse outro vetor;
  - Usualmente matrizes possuem 2 dimensões.

Matrizes

- Sintaxe

  tipo identificador[dimensao1][dimensao2];

- Em que:
  - **tipo**: é um tipo da linguagem C++;
  - **identificador**: é um identificador válido em C++;
  - **dimensao1**: é o número de linhas da matriz;
  - **dimensao2**: é o número de colunas da matriz;

Matrizes

- Assim como em vetores, não é feita nenhuma checagem de limites quanto aos índices de uma matriz
  - Embora não existam índices negativos, você pode tentar acessá-los;
  - Cuidado com o acesso indevido à memória!
- Os índices começam sempre de **zero**
  - E terminam em **dimensão-1**.

Matrizes

```
int matriz[5][5];
```

- O primeiro índice corresponde à linha;
- O segundo índice corresponde à coluna;
- Não necessariamente uma matriz é quadrada.

	0	1	2	3	4
0	1	3	5	23	88
1	34	6	54	8	5
2	25	83	6	72	4
3	27	76	53	50	90
4	92	68	70	74	27

Matrizes

- Uma matriz pode ser inicializada com conteúdo pré-definido;
- Existem duas maneiras:
  - Separar as linhas em grupos, usando chaves;
  - Sequencialmente, sem nenhuma separação.

Matrizes

```
int main()
{
    int matriz[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int matriz2[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

    return 0;
}
```

Matrizes

- Usamos dois índices para identificar cada um dos elementos de uma matriz;
- O acesso pode ser individual tanto para escrita quanto para leitura;
- Para ler ou escrever uma matriz inteira, usamos laços aninhados
  - Índice *i* para linhas;
  - Índice *j* para colunas.
- No exemplo a seguir, percorremos cada coluna de cada linha, uma por vez.

```
//leitura de uma matriz 3x3 completa
for (int i = 0; i < 3; i++)
   for (int j = 0; j < 3; j++)
      cin >> matriz[i][j];

//impressão de uma matriz 3x3 completa
for (int i = 0; i < 3; i++)
   for (int j = 0; j < 3; j++)
      cout << matriz[i][j];

//exemplo de acesso direto
cin >> matriz[2][2];
cout << matriz[2][2];
```

Matrizes

- Podemos empregar diversos tipos de percursos em matrizes, simplesmente manipulando os índices *i* e *j*:
  - Percorrer linha por linha;
  - Percorrer coluna por coluna;
  - Percorrer somente os elementos da diagonal principal;
  - Percorrer somente os elementos abaixo da diagonal principal (meia banda esquerda);
  - Percorrer somente os elementos acima da diagonal principal (meia banda direita);
  - Etc.

Matrizes

- Da mesma forma como ocorre com vetores, podemos especificar uma posição da matriz e utilizar o valor armazenado em expressões lógicas ou algébricas;
- Notem que não é possível realizar nenhuma operação sobre a matriz inteira de uma só vez
  - As operações são realizadas elemento por elemento.

```
#define TAMANHO 5
int main()
{
    int numeros[TAMANHO][TAMANHO];

    numeros[0][0]+=numeros[1][0];

    cout<<numeros[3][2]<<endl;

       numeros[3][1]=numeros[3][2]*numeros[3][3];

      if(numeros[0][3]>=numeros[4][0])
        numeros[3][0]++;
    return 0;
}
```

Mandamentos do Uso de Matrizes

1. Não alocarás dinamicamente;
2. Declararás a matriz com tamanhos **maiores** do que os limites máximos do problema;
3. Limparás a matriz antes de usar;
4. Não cobiçarás a memória do próximo.

Problemas Selecionados

- http://br.spoj.com/problems/MINHOCA/
- http://br.spoj.com/problems/MATRIZ2/
- http://br.spoj.com/problems/TRILHAS/
- http://br.spoj.com/problems/OLIMPJ09/

Um Problema de Lógica

Masmorra ou Torre?

- Em um reino, o rei uma vez por mês saía as ruas e interpelava um cidadão escolhido ao acaso, para que este fizesse alguma afirmação
  - Caso a afirmação fosse verdadeira, o cidadão seria preso em uma masmorra;
  - Caso a afirmação fosse falsa, o cidadão seria preso em uma torre.
- Certa vez, um cidadão afirmou algo ao rei, que nada pôde fazer a não ser deixá-lo livre
  - O que foi dito?

Perguntas?