Departamento de Sistemas de Computação
Universidade de São Paulo
Introdução à Ciência da Computação

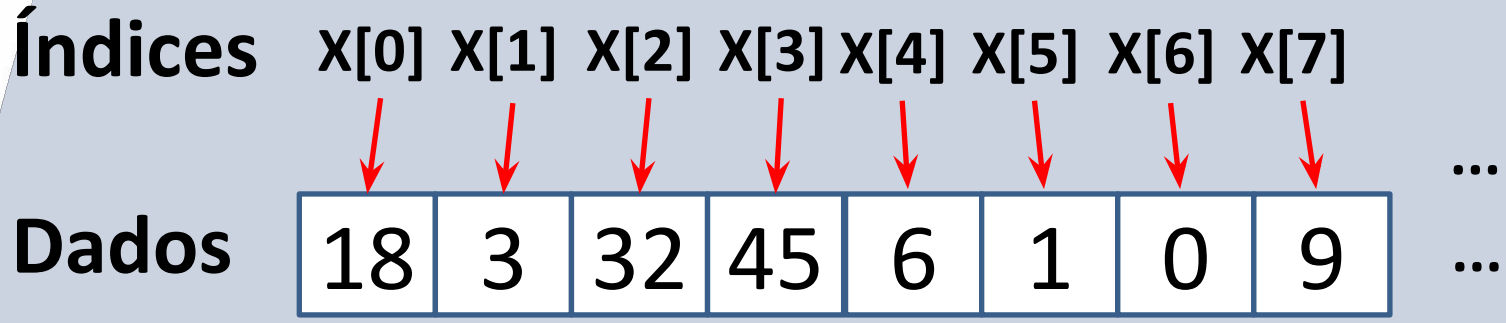
Aula 5
Matrizes

Responsável

Prof. Armando Toda (armando.toda@usp.br)

Relembrando Vetores

Vetores (Arrays)

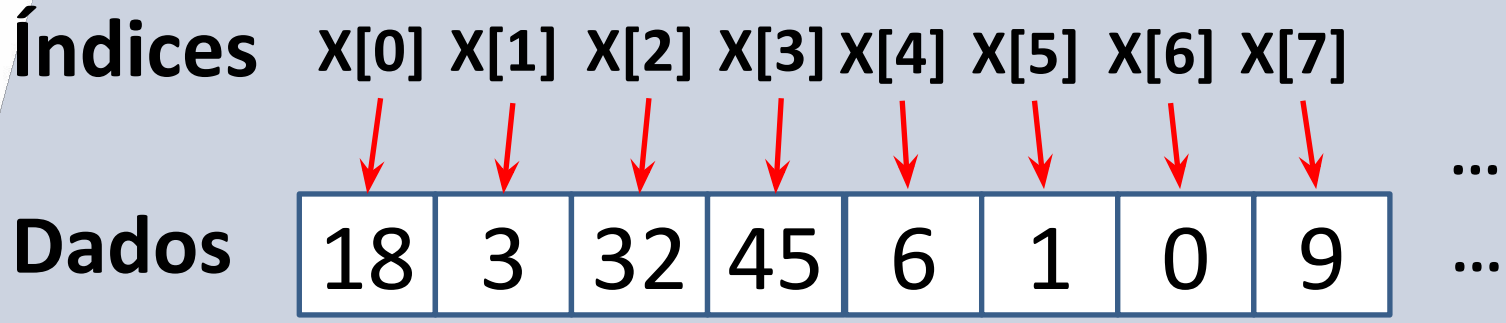


Vetores

são estruturas de dados que armazenam diversos valores de um mesmo tipo

- Se acessar o índice x[2] o dado retornado será 32

Vetores (Arrays)



Declaração

1) <tipo> <nome>[tamanho]

Vetor com um tamanho fixo

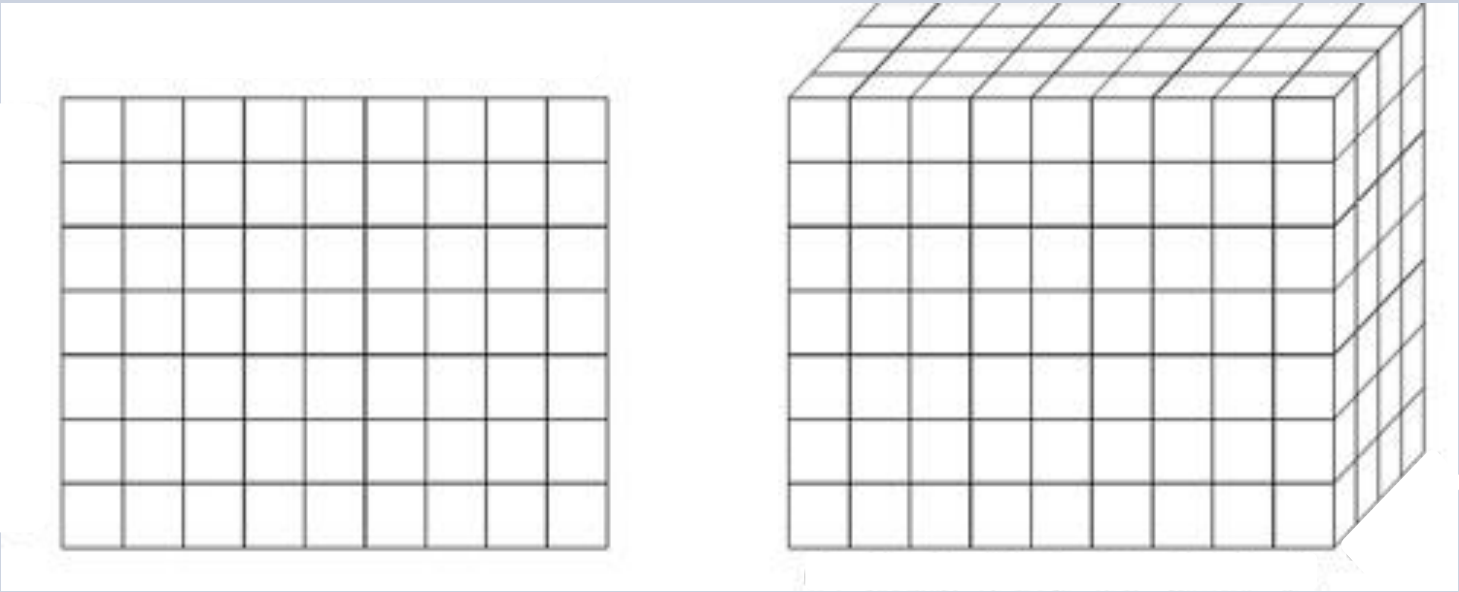
2) <tipo> <nome>[]

Vetor de tamanho qualquer

Matrizes

Matrizes

- Estrutura de dados composta, homogênea e multidimensional.
- Matrizes podem ter 2, 3, ... n dimensões.
  - M[1][3], M[0][0][0], M[0][1][1], ...



Aplicação de Matrizes (1)

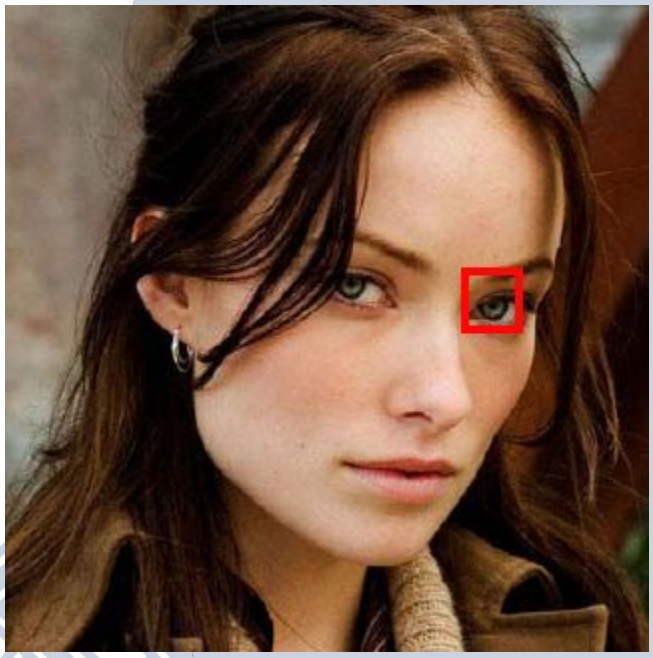
- Matrizes podem ser usadas para simular tabelas.
  - Tabelas de valores

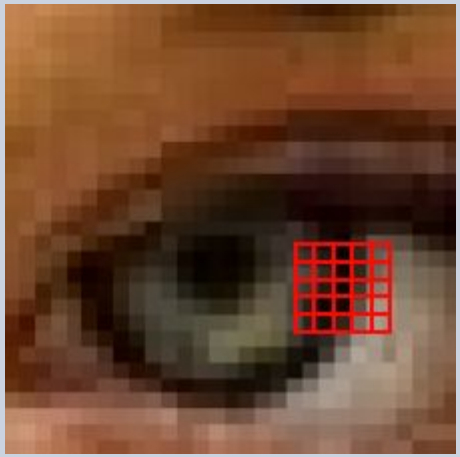
Valores em Reais das Diárias de um Hotel

	Suíte Executiva	Suíte Presidencial
Baixa-temporada	60,00	90,00
Alta-temporada	95,00	130,00
Carnaval	100,00	140,00
Natal/Ano Novo	110,00	150,00

Aplicação de Matrizes (2)

- Imagens são matrizes de pixels





Cada cor é representada por um número

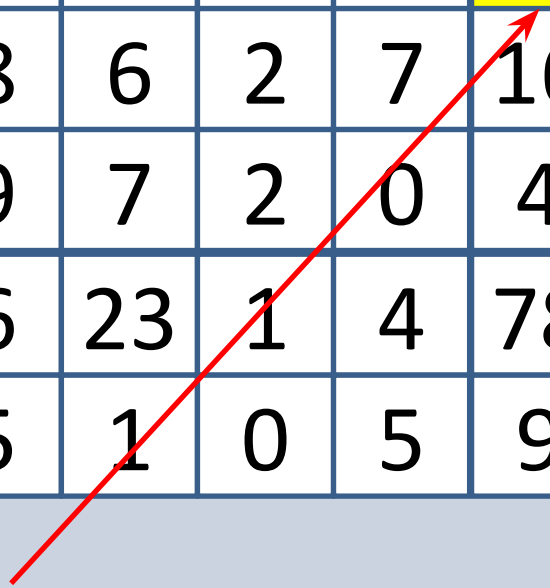
100	105	105	100	80
160	160	155	100	80
180	155	155	90	75
180	155	155	90	75
200	180	155	90	75

Matrizes

- Formada por uma sequência de variáveis do mesmo tipo, com o mesmo identificador (mesmo nome) e alocadas sequencialmente na memória.
- As variáveis são distinguidas pelos índices que referenciam sua localização dentro da estrutura.
- Há um índice para cada uma das dimensões da matriz.

Matrizes

Índices	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	…
[0]	18	32	0	15	0	4	7	2	…
[1]	1	43	4	6	61	2	4	09	…
[2]	8	6	2	7	16	1	0	56	…
[3]	9	7	2	0	4	5	3	6	…
[4]	6	23	1	4	78	34	1	9	…
[5]	5	1	0	5	9	2	7	8	…

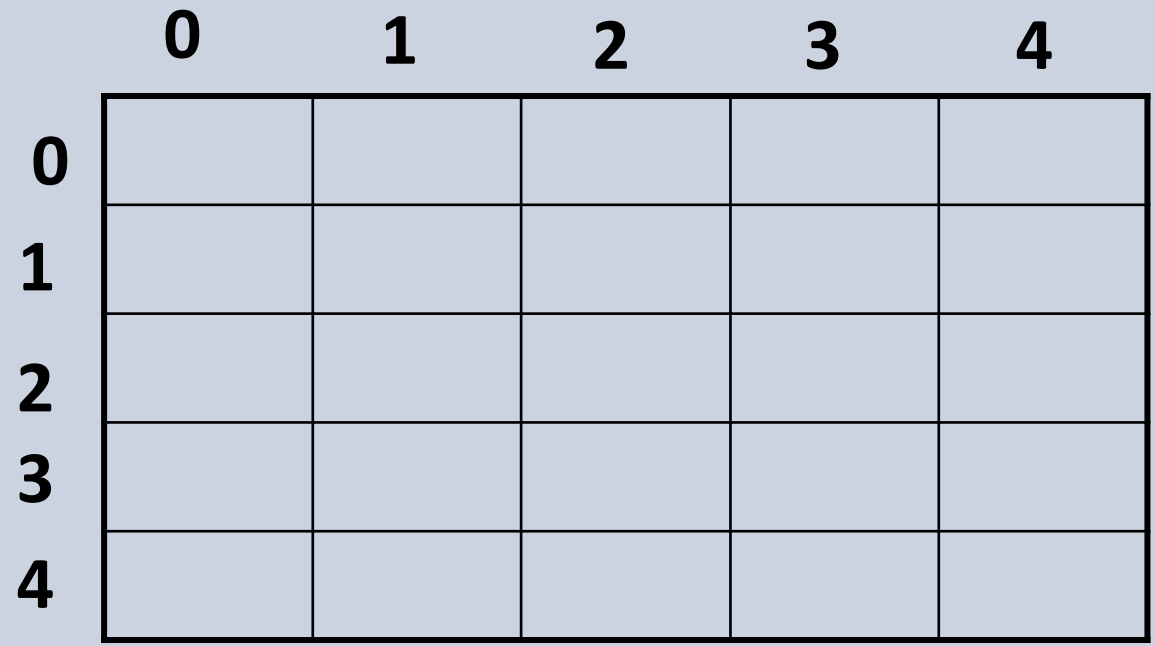


Matriz[1][4]

Matrizes - Declaração

- Na memória serão alocadas 5 x 5 posições

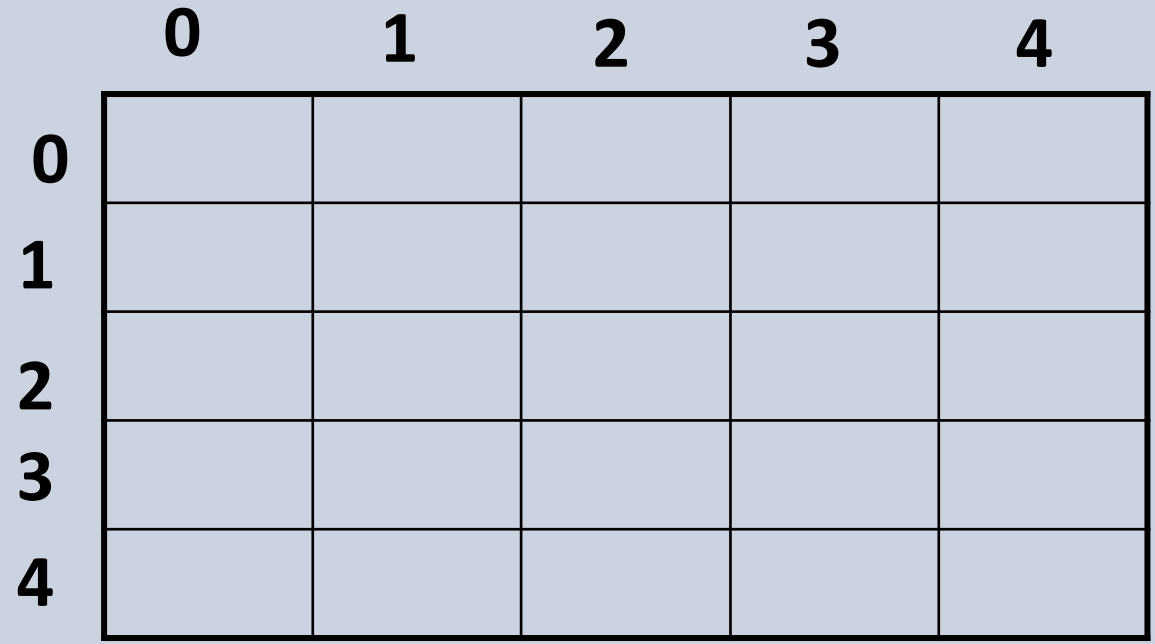
```
1) <tipo> <nome>[tamanho_linha][tamanho_coluna]
```



Matrizes - Declaração

- Na memória serão alocadas 5 x 5 posições

```
1) char mat_letras[5][5];
```



Matrizes - Acesso

- Para acessar um valor da matriz, é preciso informar 2 posições

```
mat_letras[0][0] = 'A';
```

	0	1	2	3	4
0	A				
1					
2					
3					
4					

Matrizes - Acesso

- Que resultado gera esta sequência de comandos?

```
printf("%c",mat_letras[3][1]);
printf("%c",mat_letras [0][0]);
printf("%c",mat_letras[1][3]);
printf("%c",mat_letras[2][4]);
printf("%c",mat_letras[4][4]);
printf("%c",mat_letras[0][1]);
```

	0	1	2	3	4
0	A	Z			
1				T	
2					R
3		M			
4					I

Saída na tela : MATRIZ

“Varrendo” uma linha da Matriz

```
lin = 0

for(col = 0; col < 5; col++){
    printf("%c",valor[lin][col]);
}
```

Tela

```
Valor: A
Valor: Z
Valor: ?
Valor: 3
Valor: y
```

Valor de ‘**col**’

```
0
1
2
3
4
```

	0	1	2	3	4
0	A	Z		3	Y
1		6		T	F
2	5	D	V	1	R
3	3			A	5
4	6	J	K	X	I

“Varrendo” uma coluna da Matriz

```
col = 1

for(lin = 0; lin < 5; lin++){
    printf("%c",valor[lin][col]);
}
```

Tela

```
Valor: Z
Valor: 6
Valor: D
Valor: ?
Valor: J
```

Valor de ‘**lin**’

```
0
1
2
3
4
```

	0	**1**	**2**	**3**	**4**
0	A	Z		3	Y
1		6		T	F
2	5	D	V	1	R
3	3			A	5
4	6	J	K	X	I

Matrizes - Inicialização

Inicialização de uma matriz de duas dimensões:

- Todas as posições da matriz devem ser identificadas.

```
int x[3][5];
para i = 0  até 2  passo 1 faça
    para j = 0  até 4 passo 1 faça
        mostre(“Digite o valor de x[",i,"][",j,"]”)
        leia(x[i][j]) // também pode usar x[i,j]
    fimpara
fimpara
```

Matrizes - Inicialização

Inicialização de uma matriz de duas dimensões:

- Todas as posições da matriz devem ser identificadas.

```
int x[3][5];
para i = 0  até 2  passo 1 faça
    para j = 0  até 4 passo 1 faça
        mostre(x[i][j])
    fimpara
    mostre("\n")
fimpara
```

Exercício

1) Faça um programa que leia os elementos de uma matriz inteira 5x5 e imprima apenas os elementos da diagonal principal.

R:

```
INTEIRO M[5][5]
PARA i ← 0 ATE 4 PASSO 1 FAÇA //Até == Menor igual (<=)
        PARA j ← 0 ATE 4 PASSO 1 FAÇA
                LEIA (M[i,j])
        FIMPARA
FIMPARA

PARA i ← 0 ATE 4 PASSO 1 FAÇA
        PARA j ← 0 ATE 4 PASSO 1 FAÇA
                SE i = j ENTAO
                        ESCREVA  “Posição[“ i “][“ j “] = ”M[i,j]
                FIMSE
        FIMPARA
FIMPARA
```

R:

```
DECLARE M[5][5]
PARA i ← 0 ATE 4 PASSO 1 FAÇA
        PARA j ← 0 ATE y-1 PASSO 1 FAÇA
                LEIA (M[i,j])
        FIMPARA
FIMPARA

PARA i ← 0 ATE 4 PASSO 1 FAÇA
        ESCREVA  “Posição[“ i “][“ i “] = ”M[i][i]
FIMPARA
```

Exercício

2) Faça um código que, dada uma matriz de inteiros A_{5x5}, verifique se existem elementos repetidos em A.

Exercícios

R:

```
Inteiro matriz[5][5], valorDeComp
PARA i ← 0 ATE 4 PASSO 1 FAÇA
    PARA j ← 0 ATE 4 PASSO 1 FAÇA
        valorDeComp ← A[i][j]
        PARA k ← 0 ATE 4  PASSO 1 FAÇA
            PARA l ← 0 ATE 4 PASSO 1 FAÇA
                SE (i != k) E (j != l) ENTAO
                    SE (valordeComp = A[k][l])
                    ESCREVA “repetido”
                    PARE // Break
                    FIMSE
                FIMSE
            FIMPARA
        FIMPARA
    FIMPARA
FIMPARA
```

```
#include <stdio.h>
#include <stdlib.h>
int main(){
   int A[5][5], valorDeComp;
   repetido = 0;

   for(int i = 0; i < 5; i++){
    for(int j = 0; j < 5; j++){
      A[i][j] = rand()%100; //preenche a matriz com valores aleatórios
    }
   }
```

```
for(int i = 0; i < 5; i++){
 for(int j = 0; j < 5; j++){
      valorDeComp = A[i][j];
       for(int k = 0; k < 5; k++){
              for(int l=0; l < 5; l++){
                 if(i!=k && j!=l){
                       if(valorDeComp = A[k][l]){
                          printf("A[%d][%d] Repetido\n", k,l);
                          //break;
                          }
                 }
              }
      }
 }
}
```

Exercícios

3) Escreva um programa que encontre o maior número de uma matriz bidimensional, x (linhas) por y (colunas), e mostre as suas respectivas posições.

PS: Considere que o valor das linhas e colunas é inserido pelo usuário.

R:

```
inteiro iMaior, jMaior, maiorValor, i, j;
leia(x,y);
inteiro M[x][y];
 < PreencheuAMatrizAqui >
maiorValor ← M[0][0]
iMaior ← 0
jMaior ← 0
PARA i ← 0 ATE x-1 PASSO 1 FAÇA
      PARA j ← 0 ATE y-1 PASSO 1 FAÇA
            SE maiorValor < M[i][j] ENTAO
                  maiorValor ← M[i][j]
                  iMaior ← i
                  jMaior ← j
            FIMSE
            FIMPARA
FIMPARA
IMPRIMA “Maior dos valores = maiorValor ,Indice x = iMaior,  Indice y = jMaior”
```

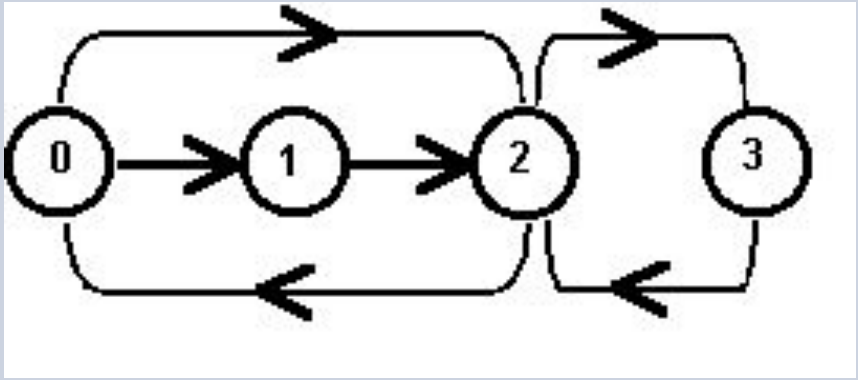
Exercícios

4) Faça um algoritmo que leia os elementos de uma matriz de inteiros 4x7 e imprima apenas os elementos que forem par.

DESAFIO EXTRA

- Considere *n* cidades numeradas de 0 a *n*-1 que estão interligadas por uma série de estradas de mão única. As ligações entre as cidades são representadas pelos elementos de uma matriz quadrada L_{nxn}, cujos elementos l_{ij} assumem o valor 1 ou 0, conforme exista ou não estrada direta que saia da cidade *i* e chegue à cidade *j*. Assim, os elementos da linha *i* indicam as estradas que saem da cidade *i*, e os elementos da coluna *j* indicam as estradas que chegam à cidade *j*.
- Por convenção $l_{ii} = 1$.

$$L = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$



(a) Dado *k*, determinar quantas estradas saem e quantas chegam à cidade *k*.

(b) A qual das cidades chega o maior número de estradas?

(c) Dado *k*, verificar se todas as ligações diretas entre a cidade *k* e outras são de mão dupla.

(d) Relacionar, se existirem:

- i. As cidades isoladas, isto é, as que não têm ligação com nenhuma outra;
- ii. As cidades das quais não há saída, apesar de haver entrada;
- iii. As cidades das quais há saída sem haver entrada.