

INF100 – Introdução à Programação I
Roteiro da Aula Prática 09 - 16 a 20 de novembro de 2020
Arranjos bidimensionais
Valor: 2 pontos


Instruções


Nome do arquivo a ser entregue: **p09.py**

Importante: Como qualquer outra prática de INF100 você deve:

1. Criar o cabeçalho obrigatório.
2. Após finalizar o cabeçalho salve o arquivo com o nome correto
3. Leia as instruções até o final e, após finalizar sua leitura, inicie sua programação.

Obs.: Recomenda-se salvar o arquivo com certa frequência para não perder a digitação já feita em caso de uma falha na rede elétrica.

 A saída do programa deve obedecer à formatação **exata** mostrada nos exemplos.

 Não esqueça de preencher o cabeçalho com seus dados e uma breve descrição do programa.

Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p09.py**) através do sistema do LBI.

Questões a serem Resolvidas

- 1) Soma de matrizes: como sabemos, a soma de duas matrizes só pode ser feita quando o número de linhas e colunas de A é igual ao número de colunas e linhas de B, sendo o algoritmo de soma implementado com o seguinte trecho de código:

```
for i in range ( 0, m ) :      # m = número de linhas de A, B e C
    for j in range ( 0, n ) :  # n = número de colunas A, B e C
        C[i][j] = A[i][j] + B[i][j]
```

Exemplo:

$$A = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 4 \\ 7 & -1 \end{bmatrix} \quad C = A + B = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} + \begin{bmatrix} 1 & 4 \\ 7 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 9 & 2 \end{bmatrix}$$

Como a soma é feita com os elementos que ocupam a mesma posição (linha, coluna), as dimensões tem que ser as mesmas.

Usando programação podemos definir duas operações de soma para matrizes de dimensões diferentes: *add-and-cut* (soma e corta) e *add-and-keep* (soma e mantém). Estas operações podem ser úteis em processamento de imagem de áreas retangulares, mas sem superposição exata.

No primeiro caso (*add-and-cut* - soma e corta), a soma vai gerar uma matriz com a menor das dimensões, eliminando os elementos que não puderem ser somados pela soma convencional (por não existir o elemento da posição correspondente na outra matriz).

Exemplo:

$$A = \begin{bmatrix} 0 & 1 & 3 \\ 2 & 3 & 1 \\ 2 & 3 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 7 & -1 & 3 & 2 \end{bmatrix} \quad C = A \text{ } aac \text{ } B = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 5 & 6 \\ 9 & 2 & 4 \end{bmatrix}$$

Em resumo, a operação soma e corta, soma as posições onde A e B tem elementos e elimina as demais posições. Considerando que m é o padrão adotado para indicar número de linhas e n o número de colunas, mA = número de linhas da matriz A, mB = número de linhas da matriz B, nA = número de colunas da matriz A, nB = número de colunas da matriz B.

Logo, a matriz resultante C terá dimensões mC (linhas de C) e nC (colunas de C) onde mC = mínimo(mA, mB) e nC = mínimo(nA, nB).

Já a operação (*add-and-keep* - soma e mantém), a soma vai gerar uma matriz com a maior das dimensões, mantendo os elementos que não puderem ser somados pela soma convencional (por não existir o elemento da posição correspondente na outra matriz) e atribuindo o valor zero para as posições que não existirem nas duas matrizes.

Exemplo:

$$A = \begin{bmatrix} 0 & 1 & 3 \\ 2 & 3 & 1 \\ 2 & 3 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 7 & -1 & 3 & 2 \end{bmatrix} \quad C = A \text{ } aac \text{ } B = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 5 & 6 & 2 \\ 9 & 2 & 4 & 2 \\ 2 & 3 & 1 & 0 \end{bmatrix}$$

Faça um programa para “somar” duas matrizes A e B, de dimensões quaisquer usando as operações *add-and-cut* e *add-and-keep*, conforme definidas acima.

O usuário deverá informar as dimensões (mA, nA) da matriz A e as dimensões (mB, nB) da matriz B. O valor mínimo para o número de linhas ou colunas deve ser 2 e o valor máximo para número de linhas ou colunas deve ser 10 (o programa deverá fazer esta validação das entradas).

O programa deverá gerar as duas matrizes A e B de números inteiros aleatórios no intervalo de 0 a 10 (esta parte já está pronta no arquivo p09.py disponível no LBI).

Na sequência o programa deverá gerar uma matriz para receber o resultado da operação add-and-cut e imprimir esta terceira matriz.

Em seguida o programa deverá gerar uma matriz para receber o resultado da operação add-and-keep e imprimir esta quarta matriz.

A figura a seguir mostra um exemplo de execução do programa:

```
Número de linhas da Matriz A (2-10): 12
Valor inválido!!!
Número de linhas da Matriz A (2-10): 2
Número de colunas da Matriz A (2-10): 8
Número de linhas da Matriz B (2-10): 3
Número de colunas da Matriz B (2-10): 4
Matriz A:
  5  0  3  3  7  9  3  5
  2  4  7  6  8  8 10  1

Matriz B:
  6  7  7  8
  1  5  9  8
  9  4  3  0

A add and cut B:
11  7 10 11
 3  9 16 14

A add and keep B:
11  7 10 11  7  9  3  5
 3  9 16 14  8  8 10  1
 9  4  3  0  0  0  0  0
```

- 2) Faça um programa para gerar uma matriz de dimensão $m \times m$ (matriz quadrada) de números aleatórios de ponto flutuante no intervalo 0 a 10. A criação da matriz já está pronta no código da questão.

O programa deve solicitar ao usuário a dimensão da matriz e aceitar valores no intervalo 3 a 8.

Na sequência o programa deverá imprimir matrizes derivadas desta matriz, para que um usuário tenha exemplos de: matriz triangular superior; matriz triangular superior e matriz diagonal.

Matriz diagonal superior é uma matriz quadrada onde todos os elementos abaixo da diagonal principal são iguais a zero.

Matriz diagonal inferior é uma matriz quadrada onde todos os elementos acima da diagonal principal são iguais a zero.

Matriz diagonal é uma matriz onde todos os elementos que não pertencem à diagonal principal são iguais a zero.

As figuras, a seguir, mostram 2 execuções do programa. **Execute o programa com a janela do Shell com largura de pelo menos 90 colunas!!!**

```
>>> q2()
Entre com a dimensão da matriz quadrada (3-8): 3

Matriz A:
0.2 2.6 1.8
8.8 1.3 2.2
4.9 7.5 5.0

Matriz Diagonal Superior:
0.2 2.6 1.8
0.0 1.3 2.2
0.0 0.0 5.0

Matriz Diagonal Inferior:
0.2 0.0 0.0
8.8 1.3 0.0
4.9 7.5 5.0

Matriz Diagonal:
0.2 0.0 0.0
0.0 1.3 0.0
0.0 0.0 5.0

>>>
```

Dica: deixe a janela do shell mais larga e imprima a parte direita da tela (Matriz Diagonal Superior e Matriz Diagonal) a partir de uma distância fixa. Em Python é possível imprimir um espaço variável parametrizado, fazendo operações aritméticas com strings. Ex:

```
esp = ' ' # 1 espaço
print( '%sOi...' %(esp*10) ) # imprime Oi... precedido de 10 espaços
```

```
>>> q2()
Entre com a dimensão da matriz quadrada (3-8): 9
Valor inválido!!!
Entre com a dimensão da matriz quadrada (3-8): 8


Matriz A:
6.3 2.0 2.9 1.8 3.8 8.3 8.4 9.2
2.2 3.6 4.8 7.4 0.7 8.6 2.0 5.7
9.9 9.5 4.8 1.5 7.6 9.2 0.8 7.7
7.3 8.8 5.6 4.3 7.3 8.9 5.1 5.3
1.5 0.9 2.8 0.7 1.6 9.3 4.2 9.5
8.0 2.0 5.7 0.7 9.0 2.6 0.2 5.7
1.7 3.6 9.0 9.7 3.2 0.2 8.6 9.1
8.4 0.4 3.2 7.6 1.3 0.6 4.0 1.4

Matriz Diagonal Superior:
6.3 2.0 2.9 1.8 3.8 8.3 8.4 9.2
0.0 3.6 4.8 7.4 0.7 8.6 2.0 5.7
0.0 0.0 4.8 1.5 7.6 9.2 0.8 7.7
0.0 0.0 0.0 4.3 7.3 8.9 5.1 5.3
0.0 0.0 0.0 0.0 1.6 9.3 4.2 9.5
0.0 0.0 0.0 0.0 0.0 2.6 0.2 5.7
0.0 0.0 0.0 0.0 0.0 0.0 8.6 9.1
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.4

Matriz Diagonal Inferior:
6.3 0.0 0.0 0.0 0.0 0.0 0.0 0.0
2.2 3.6 0.0 0.0 0.0 0.0 0.0 0.0
9.9 9.5 4.8 0.0 0.0 0.0 0.0 0.0
7.3 8.8 5.6 4.3 0.0 0.0 0.0 0.0
1.5 0.9 2.8 0.7 1.6 0.0 0.0 0.0
8.0 2.0 5.7 0.7 9.0 2.6 0.0 0.0
1.7 3.6 9.0 9.7 3.2 0.2 8.6 0.0
8.4 0.4 3.2 7.6 1.3 0.6 4.0 1.4

Matriz Diagonal:
6.3 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 3.6 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 4.8 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 4.3 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 1.6 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 2.6 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 8.6 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.4

>>>
```

 A saída do seu programa (em ambas as questões) deve obedecer à formatação **exata** mostrada nos exemplos (certifique-se que a largura da janela do Shell será suficiente para a exibição correta).

Após certificar-se que seu programa está correto, envie o arquivo do programa fonte (**p09.py**) através do sistema do LBI.

A entrega deverá ser feita até às 23h59 do dia 20/11/2020 (6ª. Feira) - Excepcionalmente nesta semana!!!