

Roteiro de Prática

Arquivo a entregar: **p10.py**

Faça o “download” do arquivo **p10.py**. Em seguida, entre no IDLE e abra o arquivo **p10.py** usando o menu **File** -> **Open** (geralmente o arquivo baixado vai para a pasta *Downloads*).

O programa **p10.py** pede um valor inteiro N, no intervalo 2 a 10, para ser usado para criar uma matriz quadrada de ordem N, ou seja, uma matriz com N linhas e N colunas.

Antes de fazer qualquer coisa, execute o arquivo **p10.py** e verifique que ele gera a seguinte saída, quando o valor 8 é fornecido.:

```
>>>
===== RESTART: /home/goulart/Do
Informe a dimensão da matriz [2, 10]: 8

Arranjo A:
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
>>> |
```

O arranjo bidimensional foi criado usando a função zeros.

Primeira tarefa:

Implemente um código para percorrer o arranjo bidimensional e atribuir um valor para cada elemento da matriz, com a seguinte lógica:

- o parte inteira do valor deverá ser igual ao número da linha na qual o valor está localizado;
- o parte fracionária do número deverá ter o valor do número da coluna na qual o valor está localizado.

Por exemplo, o valor 2.5 deve ser atribuído para o elemento da linha 2 (terceira linha) coluna 5 (sexta coluna). lembre-se que a primeira linha é a linha 0 e que a primeira coluna é a coluna 0.

Este primeiro trecho de código deve ser escrito entre as linhas 19 e 27.

Após completar esta primeira tarefa, a saída gerada pelo programa deverá ser como mostrado nos exemplos a seguir para N = 4 e N = 8:

```
>>>
===== RESTART: /hor
Informe a dimensão da matriz [2, 10]: 4

Arranjo A:
  0.0  0.1  0.2  0.3
  1.0  1.1  1.2  1.3
  2.0  2.1  2.2  2.3
  3.0  3.1  3.2  3.3
>>> |
```

```
>>>
===== RESTART: /home/goular
Informe a dimensão da matriz [2, 10]: 8

Arranjo A:
 0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7
 1.0  1.1  1.2  1.3  1.4  1.5  1.6  1.7
 2.0  2.1  2.2  2.3  2.4  2.5  2.6  2.7
 3.0  3.1  3.2  3.3  3.4  3.5  3.6  3.7
 4.0  4.1  4.2  4.3  4.4  4.5  4.6  4.7
 5.0  5.1  5.2  5.3  5.4  5.5  5.6  5.7
 6.0  6.1  6.2  6.3  6.4  6.5  6.6  6.7
 7.0  7.1  7.2  7.3  7.4  7.5  7.6  7.7
>>>
```

OBS: NÃO altere o trecho de código das linhas 29 a 33 que faz a impressão do arranjo bidimensional.

Segunda Tarefa:

A segunda tarefa consiste em criar um arranjo bidimensional B, de números inteiros, de mesmas dimensões do Arranjo A.

Dica: não se esqueça de usar **dtype=int**.

Na matemática, uma matriz com mesmo número de linhas e de colunas é chamada de matriz quadrada. Uma matriz quadrada possui uma diagonal principal formada pelos elementos em cuja posição o índice de linha é igual ao índice de coluna. Por exemplo, no arranjo bidimensional de ordem 8, os 8 elementos que estão na diagonal principal são: 0.0, 1.1, 2.2, 3.3, 4.4, 5.5, 6.6 e 7.7. E no arranjo bidimensional de ordem 4, os 4 elementos da diagonal principal são: 0.0, 1.1, 2.2 e 3.3.

Tendo uma diagonal principal, a matriz passa a ter 3 regiões:

- 1) A diagonal principal;
- 2) Os elementos que ficam acima da diagonal principal, e;
- 3) Os elementos que ficam abaixo da diagonal principal.

A seguir são descritas as propriedades de cada região, exemplificando para as posições (índices de linha e de coluna) para a matriz 4x4:

- **Diagonal principal**, onde o índice de linha é **igual** ao índice de coluna. Posições 0.0, 1.1, 2.2 e 3.3.

- **Acima da diagonal principal**, nos quais, o índice de linha é **menor** do que o índice de coluna. Posições: 0.1, 0.2 e 0.3 (na primeira linha), 1.2 e 1.3 (na segunda linha) e 2.3 (na terceira linha);

- **Abaixo da diagonal principal**, nos quais o índice de linha é **maior** do que o índice de coluna.

Escreva um trecho de código para percorrer o arranjo B e atribuir valor para cada posição de acordo com a seguinte lógica:

se “elemento está na diagonal principal”:

B(i,j) = 0 # atribuir o valor 0

senão e se “elemento está acima da diagonal principal”:

B(i, j) = 2*i + j # atribuir 2*Índice_de_linha + Índice_de_coluna

senão:

B(i, j) = i + 2*j # atribuir Índice_de_linha + 2*Índice_de_coluna

Escreva outro trecho de código, na sequência para imprimir o arranjo B.

Dica: faça uma adaptação do código das linhas 29-33, trocando o nome do arranjo (de A para B) e retirando a formatação de número de ponto flutuante “:5.1f” do comando de impressão do elemento de B (que é um valor inteiro).

Após ter completado esta segunda tarefa, a saída deverá ser como a mostrada nos exemplos a seguir (para N=4 e N=8):

```
===== RESTART: /home/g
Informe a dimensão da matriz [2, 10]: 4
```

```
Arranjo A:
0.0  0.1  0.2  0.3
1.0  1.1  1.2  1.3
2.0  2.1  2.2  2.3
3.0  3.1  3.2  3.3
```

```
Arranjo B:
0  1  2  3
1  0  4  5
2  4  0  7
3  5  7  0
```

```
>>>
```

```
===== RESTART: /home/goulart/l
Informe a dimensão da matriz [2, 10]: 8
```

```
Arranjo A:
0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7
1.0  1.1  1.2  1.3  1.4  1.5  1.6  1.7
2.0  2.1  2.2  2.3  2.4  2.5  2.6  2.7
3.0  3.1  3.2  3.3  3.4  3.5  3.6  3.7
4.0  4.1  4.2  4.3  4.4  4.5  4.6  4.7
5.0  5.1  5.2  5.3  5.4  5.5  5.6  5.7
6.0  6.1  6.2  6.3  6.4  6.5  6.6  6.7
7.0  7.1  7.2  7.3  7.4  7.5  7.6  7.7
```

```
Arranjo B:
0  1  2  3  4  5  6  7
1  0  4  5  6  7  8  9
2  4  0  7  8  9 10 11
3  5  7  0 10 11 12 13
4  6  8 10  0 13 14 15
5  7  9 11 13  0 16 17
6  8 10 12 14 16  0 19
7  9 11 13 15 17 19  0
```

```
>>> |
```

Observe que, com a lógica usada, na matriz B os valores de uma **linha x** são iguais aos valores da **coluna x** correspondente, e o valor que está **na intersecção da linha x com a coluna x é sempre 0**. Ex: a linha 2 e a coluna 2 tem os mesmos valores: 2, 4, 0, 7, 8, 9, 10 e 11.

O seu programa deve gerar uma saída exatamente como a mostrada nos exemplos acima.

Antes de enviar seu programa p10.py

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

Confira se o arquivo a ser enviado está com o nome obrigatório: [p10.py](#)

Prática 10 – INF100 – 2025/II – Valor: 2 pontos

Confira, também, se o arquivo **p10.py** contém a versão final do seu programa (verifique a data e hora que mostra a última vez em que o arquivo foi salvo).

Envie o arquivo do programa fonte (**p10.py**) através do sistema do LBI: <http://linux-server.lbi.ufv.br/>

O prazo de entrega encerra-se às 23h59 da 5a. feira, 30/10/2025.