**Target1:** Calculation of the following integral by sympy library and compare result with numpy numerical integral result.

$$\int_0^4 (x^2 + e^3 \sin(x) + 2)\, dx$$

**Description:**

1) Write the equation by using sympy library and calculate real result of it
2) Write the equation as function.
3) Create a list between boundaries which include 25 members
4) Calculate the numerical integral of the function by using numpy.trapz(y, x)
5) Print sympy output and numpy output

---

**Target 2:** Creating functions which include diagonal, upper triangular, lower triangular, band and transpose of the matrix.

**Description:**
Define 5 functions.

**1) diagonal:** It gets a matrix and output type as an input and returns diagonal of the matrix. **Use FOR loop to generate output, Do not use np.diag()** .

| Example Input | Example Output |
|---|---|
| [[0.99 0.41 0.77 0.92 0.64]<br>[0.2 0.89 0.7 0.25 0.38]<br>[0.71 0.55 0.27 0.38 0.34]<br>[0.04 0.3 0.5 0.05 0.45]<br>[0.69 0.32 0.89 0.42 0.97]] | List type:<br>[0.99, 0.89, 0.27, 0.05, 0.97]<br>Matrix type:<br>[[0.99 0. 0. 0. 0. ]<br>[0. 0.89 0. 0. 0. ]<br>[0. 0. 0.27 0. 0. ]<br>[0. 0. 0. 0.05 0. ]<br>[0. 0. 0. 0. 0.97]] |

**2) upper:** It gets a matrix as input and returns upside of the diagonal of input matrix. **Use FOR loop to generate output, Do not use np.triu()**

| Example Input | Example Output |
|---|---|
| [[0.99 0.41 0.77 0.92 0.64]<br>[0.2 0.89 0.7 0.25 0.38]<br>[0.71 0.55 0.27 0.38 0.34]<br>[0.04 0.3 0.5 0.05 0.45]<br>[0.69 0.32 0.89 0.42 0.97]] | [[0.99 0.41 0.77 0.92 0.64]<br>[0. 0.89 0.7 0.25 0.38]<br>[0. 0. 0.27 0.38 0.34]<br>[0. 0. 0. 0.05 0.45]<br>[0. 0. 0. 0. 0.97] |

**3) lower:** It gets a matrix as input and returns downside of the diagonal of input matrix. **Use FOR loop to generate output, Do not use np.tril()**

| Example Input | Example Output |
|---|---|
| [[0.99 0.41 0.77 0.92 0.64]<br>[0.2 0.89 0.7 0.25 0.38]<br>[0.71 0.55 0.27 0.38 0.34]<br>[0.04 0.3 0.5 0.05 0.45]<br>[0.69 0.32 0.89 0.42 0.97]] | [0.99 0. 0. 0. 0. ]<br>[0.2 0.89 0. 0. 0. ]<br>[0.71 0.55 0.27 0. 0. ]<br>[0.04 0.3 0.5 0.05 0. ]<br>[0.69 0.32 0.89 0.42 0.97]] |

**4) band:** It gets a matrix and bandwidth as input and returns a matrix only includes the bandwidth diagonal element of the input matrix. **Use FOR loop to generate output,**

| Example Input | Example Output |
|---|---|
| [[0.99 0.41 0.77 0.92 0.64]<br>[0.2 0.89 0.7 0.25 0.38]<br>[0.71 0.55 0.27 0.38 0.34]<br>[0.04 0.3 0.5 0.05 0.45]<br>[0.69 0.32 0.89 0.42 0.97]] | [[0.99 0.41 0. 0. 0. ]<br>[0.2 0.89 0.7 0. 0. ]<br>[0. 0.55 0.27 0.38 0. ]<br>[0. 0. 0.5 0.05 0.45]<br>[0. 0. 0. 0.42 0.97]]<br>Bandwidth:3 |

**4) Transpose:** It gets a matrix and returns transpose of the matrix. **Use FOR loop to generate output, Do not use np.transpose()**

| Example Input | Example Output |
|---|---|
| [[0.12 0.3 0.54 0.75 0.47]<br>[0.4 0.97 0.89 0.22 0.71]<br>[0.45 0.92 0.51 0.12 0.95]<br>[0.2 0.93 0.31 0.15 0.05]<br>[0.82 0.41 0.03 0.6 0.35]] | [[0.12 0.4 0.45 0.2 0.82]<br>[0.3 0.97 0.92 0.93 0.41]<br>[0.54 0.89 0.51 0.31 0.03]<br>[0.75 0.22 0.12 0.15 0.6 ]<br>[0.47 0.71 0.95 0.05 0.35]] |

**Important Note:**

- Matrixes need to be square matrix, please handle this by using try except
- Band width needs to be odd number, please handle it.
- Add comment line to the each line and put description to all functions by using “””Description“””
- Call the functions for random matrix ( np.random )