

Triangular System

Given a positive integer $n > 2$, create the matrix $A \in \mathbf{R}^{n \times n}$ such that its diagonal elements are all 2; the upper triangular part is all $\frac{1}{2}$ and the lower triangular part is all $-\frac{1}{2}$.

For example, when $n = 4$:

$$A = \begin{bmatrix} 2 & 0.5 & 0.5 & 0.5 \\ -0.5 & 2 & 0.5 & 0.5 \\ -0.5 & -0.5 & 2 & 0.5 \\ -0.5 & -0.5 & -0.5 & 2 \end{bmatrix}$$

Input Format

A positive integer $n > 2$

Output Format

A $n \times n$ numpy array representing the matrix A as described above.

Constraints

- $3 \leq n \leq 3000$

Sample Input

```
n = 4
```

Sample Output

```
[[ 2.  0.5  0.5  0.5]
 [-0.5  2.  0.5  0.5]
 [-0.5 -0.5  2.  0.5]
 [-0.5 -0.5 -0.5  2. ]]
```

Implementation

Goal: Fill in the following function:

```
def triangular_system(n):
    ...
    return ... # Return the resulting matrix
exec("\n".join(iter(input, "#Exit"))) # Don't remove this line
```