Max of Quadratic Function

Given J vectors $x_k \in \mathbb{R}^n$ for $k = 1, 2, \ldots, J$, and a symmetric matrix $A \in \mathbb{R}^{n \times n}$, compute

$$\max_{k=1,\ldots,J} x_k^T A x_k$$

The vectors x_k for $k = 1, 2, \ldots, J$ are stored as columns in a matrix $X \in \mathbb{R}^{n \times J}$.

Input Format

- Matrix $X \in \mathbb{R}^{n \times J}$ as a 2d numpy array (columns are the vectors x_k)
- Matrix $A \in \mathbb{R}^{n \times n}$ as a 2d numpy array (symmetric matrix)

Output Format

Output the maximum quadratic function value as a floating-point number.

Constraints

- $X \in \mathbb{R}^{n \times J}$ where $1 \leq n, J \leq 3000$
- $A \in \mathbb{R}^{n \times n}$ (symmetric matrix)
- All input values are real numbers in the range $[-100, 100]$

Sample Input

```
X = [[1, 2, 0, 1],
     [0, 1, 3, -1],
     [2, 0, 1, 2]]
A = [[2, 1, 0],
     [1, 3, -1],
     [0, -1, 1]]
```

Sample Output

```
22.0
```

Implementation

Goal: Fill in the following function:

```
def max_of_quadratic_function(X, A):
    ...
    return ... # Return the maximum quadratic function value
exec("\n".join(iter(input, "#Exit"))) # Don't remove this line
```