

## Maximum Inner Product

Given a matrix  $A \in \mathbb{R}^{m \times n}$ . It has  $n$  columns and each column is an  $m$ -dimensional vector as  $a_1, a_2, \dots, a_n$  where  $a_k \in \mathbb{R}^m$ . Given a vector  $x \in \mathbb{R}^m$ , find which column of  $A$  that has the maximum inner product with  $x$ , and return the inner product value.

**Definition:** The inner product between two vectors  $x$  and  $y$  is  $x^T y$ .

## Input Format

- Matrix  $A \in \mathbb{R}^{m \times n}$  as a 2d numpy array
- Vector  $x \in \mathbb{R}^m$  as a 1d numpy array

## Output Format

Output the maximum inner product value as a floating-point number.

## Constraints

- $A \in \mathbb{R}^{m \times n}$  where  $1 \leq m, n \leq 1000$
- $x \in \mathbb{R}^m$
- All input values are real numbers in the range  $[-100, 100]$

## Sample Input

```
A = [[1, 2, 3],  
      [4, 5, 6]]  
x = [1, 2]
```

## Sample Output

```
15.0
```

## Implementation

**Goal:** Fill in the following function:

```
def max_inner_product(A, x):  
    ...  
    return ... # Return the maximum inner product value  
exec("\n".join(iter(input, "#Exit"))) # Don't remove this line
```