Row-wise Softmax

Given a 2D numpy array A of shape (m, n), compute the softmax function for each row independently. The softmax function for a row $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ is defined as:

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

use `np.exp` to compute the exponential of the array.

Input Format

2D numpy array A of shape (m, n)

Output Format

return a 2D numpy array of shape (m, n) where each row sums to 1.0

Constraints

- $1 \leq m \leq 1000$ (number of rows)
- $1 \leq n \leq 1000$ (number of columns)
- $-10 \leq A_{ij} \leq 10$ (array elements)

Sample Input

```
A = np.array([[1.0, 2.0, 3.0],
              [4.0, 5.0, 6.0]])
```

Sample Output

```
np.array([[0.09003057, 0.24472847, 0.66524096],
          [0.09003057, 0.24472847, 0.66524096]])
```

Implementation

Goal: Fill in the following function:

```
import numpy as np
def softmax(A):
    ...
    return ...
if __name__ == "__main__":
    # You can test anything inside this block and can send it to grader
    # The grader will use only the function that you have implemented
    # !!! DO NOT write anything outside this block
    A = np.array([[1.0, 2.0, 3.0],
                  [4.0, 5.0, 6.0]])
    print(softmax(A))
```