Matrix exponential function via Taylor's series.
Programming Note: The following steps will handle a square matrix A of any size & any magnitude..
Step 1. Repeatedly divide A by 2 to shrink A. n=number of divisions
Step 2. Evaluate exp(A) via Taylor's series. A 20-term expansion yields ~15 significant digits, the limit of a double precision number.
Step 3. Repeatedly square the resulting exp n times
The same expm function works for a scalar x if "norme(A)" is changed to absolute value |x|.
Instructor: Nam Sun Wang

$$\exp(A) = \exp\left(2^n \cdot \frac{A}{2^n}\right) = \exp\left(\frac{A}{2^n}\right)^{2^n}$$

```
expm(A) := | "pre-processing: repeatedly divide A by 2 to shrink A"
           | n ← 0
           | while norme(A) > 1
           |   | A ← A/2
           |   | n ← n + 1
           | "Evaluate exp(A) via Taylor's series"
           | expm ← Σ_{i=0}^{20} A^i / i!
           | "post-processing: repeatedly square exp"
           | for i ∈ 1 .. n    if 0 < n
           |   expm ← expm·expm
           | return expm
```

```
expm(A) := | "pre-processing: repeatedly divide A by 2 to shrink A"
           | for n ∈ 0 .. 999
           |   | break if norme(A) ≤ 1
           |   | A ← A/2
           | "Evaluate exp(A) via Taylor's series"
           | expm ← Σ_{i=0}^{20} A^i / i!
           | "post-processing: repeatedly square exp"
           | for i ∈ 1 .. n    if 0 < n
           |   expm ← expm·expm
           | return expm
```