# Matrix Algebra Challenge (2 POINTS)

`scipy` `det` `inverse`

## Task

In this challenge, you'll need to solve a system of linear equations using the inverse matrix method in `numpy`. Linear equations arise in virtually every branch of STEM. Instead of solving it manually, let's leverage the power of linear algebra!

---

## Problem

You are given a system of linear equations in the form:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

Your task is to implement a function **`solve_linear_system`** that takes the coefficients of the system of equations as input and returns the solution using the inverse matrix method.

## Inputs:

- A matrix of coefficients $A$ for the variables $x$ and $y$.
- A vector of constants $b$ on the right-hand side.

The solution is found by solving the equation:

$$Ax' = b$$

where $x'$ is the solution vector containing the values of $x$ and $y$.

## Example

You are given the system:

$$\begin{cases} 2x + 3y = 5 \\ 4x + y = 6 \end{cases}$$

This corresponds to:

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

Here's how that looks in Python:

```
>>> A = [[2, 3], [4, 1]]
>>> b = [5, 6]
>>> solve_linear_system(A, b)
(array([1., 1.]), True)
```

In this example, the solution is $(x = 1), (y = 1)$, and the function returns a tuple with the solution and a boolean indicating if the inverse matrix exists.

## Notes:

- If the inverse of the matrix does not exist (i.e., the determinant of the matrix is zero), return `None` as the solution and `False` for the second output indicating that the system cannot be solved using the inverse matrix method.

In [ ]: