```
print("week 05!")
```

```
    week 05!
```

Ax = B: It is called the system of linear eaquation

How?

Here, A is matrix of n*n
x is a any variable
B is a vector of constants

What is the algorithm to solve for x:
x = A^(-1) *B
x = B/A

For example:

ax + by = c1
dx + fy = c2

then we want to solve using x = B/A

x + y = 5
2x + 4Y = 12
We want to find x using x = B/A

```
import numpy as np

# Define A and B matrix:
A = np.array([[1, 1], [2, 4]])
B = np.array([5, 12])

# We to fins X
X= np.linalg.solve(A, B)

print(X)
# where X is a vector, which is X= [4 1]
```

```
    [4. 1.]
```

Solve system of linear equation using python:
2x + y - z =8
-3x -y + 2z = -11
-2x + y + 2z = 3

```
# Define A and B matrix:
A = np.array([[2, 1, -1], [-3, -1, 2], [-2, 1, 2]])
B = np.array([8, -11, 3])

# We to fins X
X= np.linalg.solve(A, B)

print(X)
```

```
    [-4.  9. -7.]
```

No Solution:
Sometimes a sytem linear equations doesn't have a solution.

For example:
x + y = 2
x+ y = 3

```
# Define A and B matrix:
A = np.array([[1, 1], [1, 1]])
B = np.array([2, 3])
```

```
try:
  X= np.linalg.solve(A, B)
  print(X)
except np.linalg.LinAlgError:
  print("The systen doesn't have solution")
```

```
    The systen doesn't have solution
```

Infinitely many solution

x + y = 2

2x + 2y = 4

```
# Define A and B matrix:
A = np.array([[1, 1], [2, 2]])
B = np.array([2, 4])

try:
  X= np.linalg.solve(A, B)
  print(X)
except np.linalg.LinAlgError:
  print("The systen of equations have infinitely many solution")
```

```
    The systen of equations have infinitely many solution
```

LU decomposition:

Find L and U of the matrix A below:

A = [1 2 3

4 5 6

7 8 10]

```
import scipy.linalg

# define the matrix A
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 10]])

P, L, U = scipy.linalg.lu(A)
# P means a permutation matrix which gives by 'lu'

print("Lower matrix after decomposition:\n", L)
print("\nUpper matrix after decomposition:\n", U,)
```

```
    Lower matrix after decomposition:
     [[1.         0.         0.        ]
     [0.14285714 1.         0.        ]
     [0.57142857 0.5        1.        ]]

    Upper matrix after decomposition:
     [[ 7.          8.         10.        ]
     [ 0.          0.85714286  1.57142857]
     [ 0.          0.         -0.5       ]]
```

Solution of Systemm of Linear equation through coding:

A = [[2, 3, 1],

[4, 2, 3],

[3, 2, 2]]

B = [7, 12, 10]

Step 01: forward elimination to convert A to upper triangular form

```
# length of matrix
n = len(A)

# forward elimination
for i in range(n):
  pivot = A[i][i]
  for j in range(i + 1, n):
    factor = A[j][i] / pivot
    for k in range(i, n):
      A[j][k] -= factor * A[i][k]
    B[j-1] -= factor * B[i]

# Perform back-substitu
x = [0] * n

for i in range(n - 1, -1, -1):
  x[i] = B[i-1]

for j in range(i + 1, n):
  x[i] -= A[i][j] * x[j]
  x[i] /= A[i][i]

# Print the solution
print("Solution:")
for i in range(n):
 print(f"x[{i}] = {x[i]}")
```

```
Solution:
x[0] = -12.0
x[1] = 2
x[2] = 4
```