บทที่ 2.
เมทริกซ์
Matrix

2.1 Introduction to the Matrix

เมทริกซ์ (Matrix หรือ Matrices) เป็นรูปแบบการเก็บข้อมูลที่เหมือนเป็น “กล่อง” ของข้อมูล ที่เราสามารถเก็บข้อมูลและค้นหาข้อมูลในแนว ซ้าย-ขวา (คอลัมน์ – column) หรือ บน-ล่าง (แถว - row) ได้ โดยปกติเมทริกซ์เก็บข้อมูลเป็นตัวเลข หรือตัวแปร งานทางด้าน computer science เรารู้จักเมทริกซ์กันแล้วในรูปแบบ array 2 มิติ

ตัวอย่างการใช้งานเมทริกซ์เพื่อเก็บและแสดงข้อมูล ถ้าเราต้องการเก็บข้อมูลของเพื่อน 3 คน โดยต้องการจำนวนหน่วยกิตที่เรียนไปแล้วและเกรดเฉลี่ยที่ได้ เราสามารถเขียนในรูปแบบของเมทริกซ์ A ได้ดังนี้

$$\begin{array}{cc} & \begin{array}{ccc}\text{Jim} & \text{Jack} & \text{Joe}\end{array} \\ \begin{array}{c}\text{หน่วยกิต} \\ \text{เกรดเฉลี่ย}\end{array} & \begin{bmatrix} 20 & 22 & 21 \\ 3.20 & 2.78 & 3.45 \end{bmatrix}\end{array}$$

เมทริกซ์เป็นการเก็บข้อมูลแบบหลายมิติ (multi-dimensional) เมื่อบอกขนาดของเมทริกซ์เราต้องบอกจำนวนแถว แล้วตามด้วยจำนวนคอลัมน์เสมอ จากตัวอย่างข้อมูลของเพื่อน เมทริกซ์ A มีขนาด (dimension) เป็น 2×3 เพราะเมทริกซ์เก็บข้อมูล 2 แถว และ 3 คอลัมน์ ตัวเลขแต่ละตัวในเมทริกซ์เรียกว่าสมาชิกของเมริกซ์ (element) และสามารถเรียกสมาชิกแต่ละตัวโดยการบอกตำแหน่งแถวและคอลัมน์ เช่น เกรดเฉลี่ยของ Jim มีค่าเป็น 3.20 และคือสมาชิกตัวที่ a_{21} เพราะเป็นข้อมูลในแถวที่ 2 คอลัมน์ที่ 1

เมทริกซ์สามารถใช้ในการเก็บและนำเสนอข้อมูล และประยุกต์ใช้ในการแก้ปัญหาโจทย์สมการทางคณิตศาสตร์ แก้ปัญหาที่เราอาจเจอในชีวิตประจำวัน ซึ่งเราจะเห็นตัวอย่างในบทต่อไป

ตัวอย่างอื่น ๆ ในการใช้เมทริกซ์เพื่อแสดงข้อมูล เช่น

เปอร์เซ็นต์ยอดขาย smart phone แต่ละยี่ห้อทั่วโลก

Brands/ Years	2018	2019	2020
Huawei	14	16	18
Samsung	19	20	20
Apple	14	13	14
Xiaomi	8	8	10
Oppo	8	8	9

จำนวนนิสิตที่ได้เกรดต่าง ๆ ในแต่ละวิชา

วิชา/ เกรด	A	B	C	D	E
Programming	10	20	15	2	3
OOP	5	10	20	12	3
Database	5	5	15	20	5
Project	20	10	5	15	0

สรุปลักษณะการเขียนเมทริกซ์คือ

- เขียนสมาชิกเมทริกซ์ภายในเครื่องหมาย [...]
- ใช้ตัวอักษรพิมใหญ่ เช่น A, B, C, ...
- ขนาดของเมทริกซ์ระบุเป็น (m x n) โดย m คือจำนวนแถว และ n คือจำนวนคอลัมน์
- สมาชิกของเมทริกซ์แทนตัวพิมพ์เล็ก พร้อมทั้งระบุตำแหน่งแทนแถวและคอลัมน์ เช่น
  - a_{12} หมายถึง สมาชิกแถวที่ 1 คอลัมน์ที่ 2 ของเมทริกซ์ A
  - b_{23} หมายถึง สมาชิกแถวที่ 2 คอลัมน์ที่ 3 ของเมทริกซ์ B

ตัวอย่างการเขียนเมทริกซ์ A และสมาชิกคือ

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

ตัวอย่าง 2.1 หาขนาดและสมาชิกของเมทริกซ์ต่อไปนี้

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 0 & 4 \end{bmatrix}, B = \begin{bmatrix} 3 & 23 \\ 4 & 0 \\ 7 & 9 \end{bmatrix}, C = \begin{bmatrix} 5 \\ -8 \\ 2 \end{bmatrix}$$

วิธีทำ

เมทริกซ์ A มีขนาด	2 x 3
เมทริกซ์ B มีขนาด	3 x 2
เมทริกซ์ C มีขนาด	_______________
a_{21} มีค่าเป็น	2
b_{12} มีค่าเป็น	23
c_{31} มีค่าเป็น	_______________
c_{22} มีค่าเป็น	_______________

2.1.1 ประเภทของเมทริกซ์

เมทริกซ์ที่มีลักษณะพิเศษและมีชื่อเรียกเฉพาะ มีดังนี้

ประเภท	**คำอธิบาย**	**ตัวอย่าง**
เมทริกซ์ศูนย์ Zero matrix	เมทริกซ์ที่สมาชิกทุกตัวเป็นศูนย์ เขียนแทนด้วย $\underline{0}$	$A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
เมทริกซ์จัตุรัส Square matrix	เมทริกซ์ที่มีจำนวนแถวเท่ากับจำนวนคอลัมน์ ขนาดของเมทริกซ์เป็น n x n เขียนแทนด้วย A_n	$A_2 = \begin{bmatrix} 1 & 7 \\ 1 & 0 \end{bmatrix}$ $A_3 = \begin{bmatrix} 0 & 12 & 5 \\ 2 & 10 & 2 \\ 2 & -3 & 4 \end{bmatrix}$
เมทริกซ์เอกลักษณ์ Identity matrix	เมทริกซ์จัตุรัสที่สมาชิกในแนวทแยงมีค่าเป็นหนึ่งและ สมาชิกตำแหน่งอื่น ๆ มีค่าเป็นศูนย์ เขียนแทนด้วย $\mathbf{I}_n$	$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
เมทริกซ์แนวทแยง Diagonal matrix	เมทริกซ์จัตุรัสที่สมาชิกในแนวทแยงเป็นศูนย์	$A_2 = \begin{bmatrix} 0 & 3 \\ 0 & 0 \end{bmatrix}$ $A_3 = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 0 & 5 \\ 6 & 9 & 0 \end{bmatrix}$
เมทริกซ์สามเหลี่ยมบน Upper triangular matrix หรือ Right triangular matrix	เมทริกซ์จัตุรัสที่สมาชิกที่อยู่ใต้แนวทแยงเป็นศูนย์ สมาชิกตัวอื่นเป็นค่าอะไรก็ได้	$\begin{bmatrix} 1 & 2 & 0 & 5 \\ 0 & 4 & 0 & 3 \\ 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 7 \end{bmatrix}$
เมทริกซ์สามเหลี่ยมล่าง Lower triangular matrix หรือ Left triangular matrix	เมทริกซ์จัตุรัสที่สมาชิกที่อยู่เหนือแนวทแยงเป็นศูนย์ สมาชิกตัวอื่นเป็นค่าอะไรก็ได้	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 7 & 0 & 5 & 0 \\ 0 & 2 & 5 & 7 \end{bmatrix}$

2.1.2 การเท่ากันของเมทริกซ์ (Equality of matrices)

เมทริกซ์ A เท่ากับเมทริกซ์ B ก็ต่อเมื่อสมาชิกทุกตัวของเมทริกซ์ A มีค่าเท่ากันกับสมาชิกที่ตำแหน่งเดียวกันของเมทริกซ์ B นั่นคือ ถ้า $A = [a_{ij}]_{m\times n}$ และ $B = [b_{ij}]_{p\times q}$ แล้ว
A = B ก็ต่อเมื่อ m = p, n = q และ $a_{ij} = b_{ij}$ สำหรับทุกค่า i และ j

ตัวอย่าง 2.2 จากเมทริกซ์ที่กำหนดให้ จงหาค่า a และ b

กำหนดให้ $\begin{bmatrix} a^2 & b-3 \\ b^2 & -1 \end{bmatrix} = \begin{bmatrix} 49 & 3 \\ a+43 & a+b \end{bmatrix}$

วิธีทำ

จากข้อมูลของ 2 เมทริกซ์ที่มีค่าเท่ากัน ดังนั้น

$a^2 = 49$ ________ (1)

$b^2 = a + 43$ ________ (2)

$\mathrm{b} - 3 = 3$ ________ (3)

$\mathrm{a} + \mathrm{b} = -1$ ________ (4)

จากสมการ (3) จะได้ b = 6

แทนค่า b = 6 ในสมการ (4) จะได้ a = -7

2.1.3 การบวกและลบเมทริกซ์ (Adding and subtracting matrices)

เราสามารถบวกหรือลบเมทริกซ์ได้ ก็ต่อเมื่อเมทริกซ์ที่นำมาบวกหรือลบนั้น มีขนาดเท่ากัน (จำนวนแถวและคอลัมน์เท่ากัน) โดยสามารถบวกหรือลบสมาชิกในตำแหน่งที่ตรงกันของเมทริกซ์

ตัวอย่าง 2.3 กำหนดให้ $A = \begin{bmatrix} 2 & 4 \\ 6 & 0 \\ 1 & 3 \end{bmatrix}$ และ $B = \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 4 & 2 \end{bmatrix}$ หาค่าของ A + B และ A – B

วิธีทำ

$$A + B = \begin{bmatrix} 2 & 4 \\ 6 & 0 \\ 1 & 3 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 2+1 & 4+2 \\ 6+0 & 0+0 \\ 1+4 & 3+2 \end{bmatrix} = \begin{bmatrix} 3 & \_\_\_\_ \\ 6 & \_\_\_\_ \\ 5 & \_\_\_\_ \end{bmatrix}$$

$$A - B = \begin{bmatrix} 2 & 4 \\ 6 & 0 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 0 & 0 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 1 & \_\_\_\_ \\ 6 & \_\_\_\_ \\ -3 & \_\_\_\_ \end{bmatrix}$$

ตัวอย่าง 2.4 เราต้องการเก็บข้อมูลการซื้อของตลอด 3 เดือน โดยเก็บข้อมูลแยกเป็นซื้อที่ร้าน และซื้อออนไลน์ สิ่งของที่เราซื้อคือ เสื้อ กางเกง และหมวก เราสามารถเก็บข้อมูลเหล่านี้ในรูปแบบเมทริกซ์ได้ดังนี้

มกราคม

	ร้าน	Online
เสื้อ	2	5
กางเกง	2	2
หมวก	1	0

เมทริกซ์: $\begin{bmatrix} 2 & 5 \\ 2 & 2 \\ 1 & 0 \end{bmatrix}$

กุมภาพันธ์

	ร้าน	Online
เสื้อ	1	0
กางเกง	4	3
หมวก	2	2

เมทริกซ์: $\begin{bmatrix} 1 & 0 \\ 4 & 3 \\ 2 & 2 \end{bmatrix}$

มีนาคม

	ร้าน	Online
เสื้อ	4	2
กางเกง	5	3
หมวก	4	1

เมทริกซ์: $\begin{bmatrix} 4 & 2 \\ 5 & 3 \\ 4 & 1 \end{bmatrix}$

เมทริกซ์ทั้งสามมีขนาดเท่ากันคือ 3×2 ดังนั้นถ้าเราต้องการดูจำนวนสิ่งของที่เราซื้อทั้ง 3 เดือน เราสามารถบวกเมทริกซ์ทั้งสามได้ดังนี้

$$\begin{bmatrix} 2 & 5 \\ 2 & 2 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 4 & 3 \\ 2 & 2 \end{bmatrix} + \begin{bmatrix} 4 & 2 \\ 5 & 3 \\ 4 & 1 \end{bmatrix} = \begin{bmatrix} 2+1+4 & 5+0+2 \\ 2+4+5 & 2+3+3 \\ 1+2+4 & 0+2+1 \end{bmatrix} = \begin{bmatrix} 7 & 7 \\ 11 & 8 \\ 7 & 3 \end{bmatrix}$$

จากผลลัพธ์การบวกเมทริกซ์ เราจะเห็นข้อมูลได้ง่ายขึ้น ว่า เราซื้อเสื้อจากร้านค้า 7 ตัว ซื้อออนไลน์อีก 7 ตัว ซื้อกางเกงจากร้านค้า 11 ตัว ซื้อออนไลน์ 8 ตัว และซื้อหมวกจากร้านค้า 7 ใบ ซื้อออนไลน์ 3 ใบ

2.1.4 การคูณเมทริกซ์ด้วยสเกลาร์ (Scalar multiplication matrix)

เราสามารถคูณเมทริกซ์ด้วยค่าคงที่หรือสเกลาร์ได้ โดยการคูณค่าคงที่นั้นกับสมาชิกทุกตัวในเมทริกซ์

หมายเหตุ การคูณเมทริกซ์ด้วยสเกลาร์แตกต่างจากการคูณเมทริกซ์ด้วยเมทริกซ์ซึ่งจะอธิบายในหัวข้อต่อไป

ตัวอย่าง 2.5 กำหนดให้ $A = \begin{bmatrix} 2 & 4 \\ 6 & 0 \\ 1 & 3 \end{bmatrix}$

$$2A = \begin{bmatrix} 4 & 8 \\ 12 & 0 \\ 2 & 6 \end{bmatrix}$$

$$-5A = \begin{bmatrix} -10 & -20 \\ -30 & 0 \\ -5 & -15 \end{bmatrix}$$

ตัวอย่าง 2.6 จาก

ตัวอย่าง 2.4 ถ้าเราต้องการรู้ว่า ถ้าเดือนมีนาคมเราซื้อของเป็น 5 เท่าของที่ซื้อไปจริงๆ ในเดือนนี้เราจะซื้อของทั้งหมดกี่ชิ้น เราสามารถทำได้โดยการคูณเมทริกซ์ของเดือนมีนาคม ด้วย 5

$$5\begin{bmatrix}4 & 2\\5 & 3\\4 & 1\end{bmatrix} = \begin{bmatrix}4\times 5 & 2\times 5\\5\times 5 & 3\times 5\\4\times 5 & 1\times 5\end{bmatrix} = \begin{bmatrix}20 & 10\\25 & 15\\20 & 5\end{bmatrix}$$

นั่นคือ เดือนมีนาคมเราจะซื้อเสื้อจากร้านค้า 20 ตัว และซื้อออนไลน์ 10 ตัว (และอื่น ๆ)

2.1.5 การคูณเมทริกซ์ด้วยเมทริกซ์ (Multiplying matrices)

การคูณเมทริกซ์ด้วยเมทริกซ์ มีเทคนิคนิดหน่อย คือเราไม่สามารถจับเมทริกซ์ 2 ตัวใด ๆ ก็ได้มาคูณกันได้เลย เมทริกซ์ที่จะมาคูณกันได้ต้องมีขนาดที่ “พอดี” กันด้วย ขนาดพอดีกันที่ว่านี้คือ จำนวนคอลัมน์ของเมทริกซ์ตัวแรกต้องเท่ากับจำนวนแถวของเมทริกซตัวที่สอง ไม่เช่นนั้นจะไม่สามารถคูณเมทริกซ์สองตัวนี้ได้ นั่นคือ

เมทริกซ์ A และ B สามารถคูณกันได้ก็ต่อเมื่อ จำนวนคอลัมน์ของ A เท่ากับจำนวนแถวของ B

ผลคูณของเมทริกซ์ A และ B เขียนแทนด้วย AB

กำหนดให้ $A = [a_{ij}]_{m\times n}$ และ $B = [b_{ij}]_{n\times r}$ แล้ว ถ้า $C = AB$ ดังนั้น $C = [c_{ij}]_{m\times r}$

ข้อสังเกต เมทริกซ์ผลลัพธ์ มีขนาดเป็น m × r

รูปภาพแสดงการคูณเมทริกซ์

$$\begin{bmatrix}a_{11} & a_{12} & a_{13}\\a_{21} & a_{22} & a_{23}\\a_{31} & a_{32} & a_{33}\\a_{41} & a_{42} & a_{43}\end{bmatrix}\begin{bmatrix}b_{11} & b_{12}\\b_{21} & b_{22}\\b_{31} & b_{32}\end{bmatrix} = \begin{bmatrix}\boxed{c_{11}} & c_{12}\\c_{21} & c_{22}\\c_{31} & c_{32}\\c_{41} & c_{42}\end{bmatrix}$$

4 × 3 **3×2** **4×2**

โดยที่ $c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}$

$c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32}$

$c_{21} = a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31}$ และค่าอื่น ๆ

ตัวอย่าง 2.7 กำหนดให้ $A = \begin{bmatrix} 5 & 1 & 2 \\ 2 & 3 & 0 \end{bmatrix}$ และ $B = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$ หาค่าของ C เมื่อ $C = AB$

วิธีทำ เมทริกซ์ C มีขนาด 2×1

$$C = \begin{bmatrix} 5 & 1 & 2 \\ 2 & 3 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 5(2) + 1(0) + 2(1) \\ 2(2) + 3(0) + 0(1) \end{bmatrix} = \begin{bmatrix} 12 \\ 4 \end{bmatrix}$$

ตัวอย่าง 2.8 กำหนดให้ $A = \begin{bmatrix} 2 & 0 & -1 \\ 4 & 1 & 2 \\ -1 & 2 & 3 \\ 3 & -1 & 4 \end{bmatrix}$ และ $B = \begin{bmatrix} 3 & 2 \\ 1 & 0 \\ 5 & 1 \end{bmatrix}$ หาค่าของ C เมื่อ $C = AB$

วิธีทำ เมทริกซ์ C มีขนาด ________________

$$C = \begin{bmatrix} 2 & 0 & -1 \\ 4 & 1 & 2 \\ -1 & 2 & 3 \\ 3 & -1 & 4 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 1 & 0 \\ 5 & 1 \end{bmatrix} = \begin{bmatrix} 2(3) + 0(1) + \ -1(5) & 2(2) + 0(0) + \ -1(1) \\ 4(3) + 1(1) + 2(5) & 4(2) + 1(0) + 2(0) \\ \underline{\hspace{6cm}} & \underline{\hspace{6cm}} \\ \underline{\hspace{6cm}} & \underline{\hspace{6cm}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \underline{\hspace{1cm}} \\ 15 & \underline{\hspace{1cm}} \\ 2 & \underline{\hspace{1cm}} \\ 28 & \underline{\hspace{1cm}} \end{bmatrix}$$

ตัวอย่าง 2.9 เมทริกซ์อยู่รอบตัวเรา

เราต้องการคำนวณเกรดของนิสิต โดยคิดจากการเก็บคะแนน 4 ประเภทคือ สอบมิดเทอม สอบไฟนอล การบ้าน และโปรเจค การเก็บคะแนนแต่ละประเภทเก็บ 100 คะแนน แต่สุดท้ายต้องนำคะแนนมาคำนวณเปอร์เซ็นต์ซึ่งมีน้ำหนักไม่เท่ากัน ให้หาว่านิสิตแต่ละคนได้คะแนนรวมสุดท้ายเท่าไร

คะแนนเก็บที่นิสิตได้ดังนี้

นิสิต	Midterm exam	Final exam	Homework	Project
Jim	100	90	52	60
Jack	80	85	90	100
Joe	92	50	80	70

การเก็บคะแนน	เปอร์เซ็นต์
Midterm exam	20% (0.2)
Final exam	30% (0.3)
Homework	10% (0.1)
Project	40% (0.4)

เราสามารถนำผลคูณเมทริกซ์มาใช้ในการคำนวณคะแนนรวมของนิสิตได้ดังนี้

เมทริกซ์	ผลคูณ
$\begin{bmatrix} 100 & 90 & 52 & 60 \\ 80 & 85 & 90 & 100 \\ 92 & 50 & 80 & 70 \end{bmatrix} \times \begin{bmatrix} .2 \\ .3 \\ .1 \\ .4 \end{bmatrix} = \begin{bmatrix} 76.2 \\ 90.5 \\ 69.4 \end{bmatrix}$	(100×.2) + (90×.3) + (52×.1) + (60×.4) = 76.2 (80×.2) + (85×.3) + (90×.1) + (100×.4) = 90.5 (92×.2) + (50×.3) + (80×.1) + (70×.4) = 69.4

ผลรวมคะแนนสุดท้ายของนิสิตแต่ละคน คือ Jim ได้ 76.2 คะแนน Jack ได้ 90.5 คะแนน Joe ได้ 69.4 คะแนน

ตัวอย่าง 2.10 ตัวอย่างนี้แสดงให้เห็นว่า การคูณเมทริกซ์สามารถนำไปประยุกต์ใช้ได้อย่างไร

เราทำตุ๊กตาขาย ตุ๊กตาแต่ละสีขายในราคาที่แตกต่างกัน โดยการขายต้องขายส่งทั้งหมดให้กับลูกค้าที่รับซื้อ ราคาตุ๊กตาที่ลูกค้าแต่ละคนเสนอซื้อตามตารางต่อไปนี้

ลูกค้า / สีตุ๊กตา	แดง	ขาว	น้ำเงิน
Jim	3	2	4
Jack	4	3	1
Joe	5	1	2
Jane	4	2	3
Joy	2	1	10

เราทำตุ๊กตาสีแดง 1000 ตัว สีขาว 800 ตัว และสีน้ำเงิน 200 ตัว ถามว่าเราควรขายตุ๊กตาให้ลูกค้าคนใดจึงจะได้เงินมากที่สุด

วิธีทำ

จากข้อมูลที่มีสามารถสร้างเป็นเมทริกซ์ได้คือ

A เป็นเมทริกซ์ราคา $= \begin{bmatrix} 3 & 2 & 4 \\ 4 & 3 & 1 \\ 5 & 1 & 2 \\ 4 & 2 & 3 \\ 2 & 1 & 10 \end{bmatrix}$

B เป็นเมทริกซ์จำนวน $= \begin{bmatrix} 1000 \\ 800 \\ 200 \end{bmatrix}$

ผลคูณของเมทริกซ์ C = AB แสดงถึงรายได้ ซึ่งสมาชิกแต่ละตัวแทนรายได้จากการขายตุ๊กตาทั้ง 3 ชนิดให้ลูกค้าแต่ละคน

$$C = \begin{bmatrix} 3 & 2 & 4 \\ 4 & 3 & 1 \\ 5 & 1 & 2 \\ 4 & 2 & 3 \\ 2 & 1 & 10 \end{bmatrix} \begin{bmatrix} 1000 \\ 800 \\ 200 \end{bmatrix} = \begin{bmatrix} 3000 + 1600 + 800 \\ 4000 + 2400 + 200 \\ 5000 + 800 + 400 \\ 4000 + 1600 + 600 \\ 2000 + 800 + 2000 \end{bmatrix} = \begin{bmatrix} 5400 \\ 6600 \\ 6200 \\ 6200 \\ 4800 \end{bmatrix}$$

ลูกค้าที่รับซื้อตุ๊กตาในราคาสูงที่สุดคือซื้อในราคา 6600 บาท ดังนั้นควรขายตุ๊กตาให้ Jack

2.1.6 คุณสมบัติการบวกและคูณของเมทริกซ์

คุณสมบัติการบวกของเมทริกซ์ กำหนดให้ A, B และ C เป็นเมทริกซ์ขนาด $m \times n$	
การเปลี่ยนกลุ่ม	$(A + B) + C = A + (B + C)$
การสลับที่	$A + B = B + A$
เอกลักษณ์การบวก คือ $\underline{0}$	$A + \underline{0} = \underline{0} + A = A$
อินเวอร์สการบวก (Inverse)	อินเวอร์สการบวกของเมทริกซ์ A คือ –A ที่เมื่อนำมาบวกกับ A แล้วทำให้ได้ $\underline{0}$ $A + (-A) = (-A) + A = \underline{0}$

คุณสมบัติการคูณเมทริกซ์ด้วยค่าคงที่หรือสเกลาร์ กำหนดให้ A, B เป็นเมทริกซ์ขนาด $m \times n$ และ c, d เป็นค่าคงที่	
การเปลี่ยนกลุ่ม	$(cd) A = c (dA) = (cA) d$
เอกลักษณ์การคูณ คือ 1	$1A = A$
การกระจาย	$c(A + B) = cA + cB$ และ $(c+d)A = cA + dA$

คุณสมบัติการคูณเมทริกซ์ด้วยเมทริกซ์ กำหนดให้ A, B และ C เป็นเมทริกซ์ขนาด $n \times n$	
การเปลี่ยนกลุ่ม	(AB) C = A (BC)
เอกลักษณ์การคูณ	คือเมทริกซ์ I ขนาด $n \times n$ ที่ทำให้ A**I** = **I**A = A โดยที่ $I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$
อินเวอร์ส์การคูณ	อินเวอร์ส์การคูณของเมทริกซ์ A คือ A^{-1} ที่เมื่อนำมาคูณกับA แล้วทำให้ผลลัพธ์เป็น $\mathbf{I}_n$ $A A^{-1} = A^{-1} A = \mathbf{I}_n$ **ข้อสังเกต** มีเพียงเมทริกซ์จัตุรัสเท่านั้น ที่สามารถหาอินเวอร์สการคูณได้ และเมทริกซ์บางตัวอาจไม่มีอินเวอร์สการคูณก็ได้ การหาอินเวอร์สของเมทริกซ์ขนาด 2×2 ถ้า $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $A^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ ดูเพิ่มเติมวิธีการหาอินเวอร์สการคูณใน **ตัวอย่าง 2.12**
<u>ข้อสังเกตุ</u>	เมทริกซ์ไม่มีคุณสมบัติในการสลับที่การคูณ นั่นคือ $A \times B \neq B \times A$ และโดยทั่วไปแม้ว่าเราจะสามารถทำการคูณเมทริกซ์ $A \times B$ ได้ แต่เราอาจไม่สามารถทำ $B \times A$ ได้ เช่น พิจารณาเมทริกซ์ $A_{2\times3}$ และ $B_{3\times4}$ เราสามารถทำ $A \times B$ ได้ แต่ไม่สามารถทำ $B \times A$ ได้
คุณสมบัติของอินเวอร์สการคูณเมทริกซ์	ให้ A และ B ขนาด $n \times n$ I_n เป็นเมทริกซ์เอกลักษณ์ และ c เป็นจำนวนจริง $(A^{-1})^{-1} = A$ $(\mathrm{AB})^{-1} = A^{-1}B^{-1}$ $(A^{\mathrm{t}})^{-1} = (A^{-1})^{\mathrm{t}}$ $(\mathrm{cA})^{-1} = {}^{1}/_{c}\,(A)^{-1}$ เมื่อ $c \neq 0$ $A\mathrm{A}^{-1} = I_n = \mathrm{A}^{-1}\mathrm{A}$

ตัวอย่าง 2.11 หาเมทริกซ์เอกลักษณ์การบวกและอินเวอร์สการบวกของเมทริกซ์ A เมื่อ

$$A = \begin{bmatrix} 2 & 0 \\ 5 & -1 \\ 3 & 1 \end{bmatrix}$$

วิธีทำ

เอกลักษณ์การบวกของ A คือเมทริกซ์ $\underline{0}$ ที่เมื่อบวกกับ A แล้วทำให้ A มีค่าเท่าเดิม นั่นคือ $\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$

อินเวอร์สการบวกของ A คือเมทริกซ์ที่เมื่อบวกกับ A แล้วทำให้ได้ $\underline{0}_{3\times 2}$ นั่นคือ $\begin{bmatrix} -2 & 0 \\ -5 & 1 \\ -3 & -1 \end{bmatrix}$

ตัวอย่าง 2.12 หาเมทริกซ์เอกลักษณ์การคูณและอินเวอร์สการคูณของเมทริกซ์ A เมื่อ

$$A = \begin{bmatrix} 2 & 0 \\ 5 & -1 \end{bmatrix}$$

วิธีทำ

เอกลักษณ์การคูณของ A คือเมทริกซ์ที่คูณกับ A แล้วทำให้ A มีค่าเท่าเดิม

A มีขนาด 2 × 2 ดังนั้น เอกลักษณ์การคูณของ A คือ I_2 นั่นคือ $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

อินเวอร์สการคูณของ A คือเมทริกซ์ขนาด 2 × 2 ที่คูณกับ A แล้วทำให้ได้เมทริกซ์เอกลักษณ์ (I_2) นั่นคือ

$$A^{-1} = \frac{1}{(2 \times -1) - (0 \times 5)} \begin{bmatrix} -1 & 0 \\ -5 & 2 \end{bmatrix}$$

$$A^{-1} = -\frac{1}{2} \begin{bmatrix} -1 & 0 \\ -5 & 2 \end{bmatrix}$$

ดังนั้น $A^{-1} = \begin{bmatrix} 1/2 & 0 \\ 5/2 & -1 \end{bmatrix}$

ทดสอบคำนวณเพื่อยืนยันคำตอบ

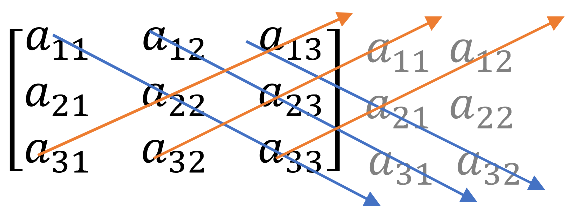
$$\begin{bmatrix} 2 & 0 \\ 5 & -1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 5/2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2.1.7 ค่ากำหนด (ดีเทอร์มิแนนต์ - Determinant)

ดีเทอร์มิแนนต์ของเมทริกซ์ คือ ฟังก์ชันที่ให้ผลลัพธ์เป็นสเกลาร์ซึ่งขึ้นอยู่กับค่าของ n ในขนาดของเมทริกซ์จัตุรัสขนาด n × n ดีเทอร์มิแนนต์ของเมทริกซ์ A เขียนแทนด้วย $\det(A)$ หรือ $|A|$

ถ้าให้ A เป็นเมทริกซ์จัตุรัสขนาด n × n

- ถ้า $\det(A) = 0$ แล้ว A เป็นเมทริกซ์เอกฐาน (Singular matrix)
- ถ้า $\det(A) \neq 0$ แล้ว A เป็นเมทริกซ์ไม่เป็นเอกฐาน (Non-singular matrix)
- ถ้า A เป็น non-singular matrix แล้ว A สามารถหาอินเวอร์สการคูณได้
- ถ้า $A = [a]$ เป็นเมทริกซ์ขนาด 1×1 และ a เป็นจำนวนจริงแล้ว $\det(A) = a$
- ถ้า $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ เป็นเมทริกซ์ขนาด 2×2 และ a, b, c, d เป็นจำนวนจริงแล้ว $\det(A) = ad - bc$
- โดยสรุป การหาค่า $\det(A)$ คือ (ผลคูณทแยงลง) - (ผลคูณทแยงขึ้น)



นั่นคือ $\det(A) =$

$$[(a_{11}a_{22}a_{33}) + (a_{12}a_{23}a_{31}) + (a_{13}a_{21}a_{32})] - [(a_{31}a_{22}a_{13}) + (a_{32}a_{23}a_{11}) + (a_{33}a_{21}a_{12})]$$

ตัวอย่าง 2.13 กำหนดเมทริกซ์ A ต่อไปนี้ ให้หา $\det(A)$ และ A^{-1}

เมทริกซ์ A	ดีเทอร์มิแนนต์ $\det(A)$	อินเวอร์สการคูณ A^{-1}
$[5]$	5	$1/5$
$\begin{bmatrix}1 & 2\\4 & 6\end{bmatrix}$	$1(6)-2(4)= -2$	$\frac{1}{1(6)-2(4)}\begin{bmatrix}6 & -2\\-4 & 1\end{bmatrix}=\begin{bmatrix}-3 & 1\\2 & -1/2\end{bmatrix}$
$\begin{bmatrix}2 & -5\\0 & 3\end{bmatrix}$	$2(3)-(-5)(0)=6$	$\frac{1}{2(3)-(-5)(0)}\begin{bmatrix}3 & 5\\0 & 2\end{bmatrix}=\begin{bmatrix}1/2 & 5/6\\0 & 1/3\end{bmatrix}$
$\begin{bmatrix}2 & 3\\4 & 6\end{bmatrix}$	$2(6)-(3)(4)=0$	$\frac{1}{2(6)-3(4)}\begin{bmatrix}6 & -3\\-4 & 2\end{bmatrix}=\frac{1}{0}\begin{bmatrix}6 & -3\\-4 & 2\end{bmatrix}$ ซึ่ง $\frac{1}{0}$ ไม่ใช่จำนวนจริง ดังนั้น ไม่สามารถหา A^{-1} ได้ ซึ่งตรงตามนิยามว่าถ้า $\det(A)=0$ จะไม่สามารถหา A^{-1} ได้

2.1.8 ทรานสโพสของเมทริกซ์ (Transpose of matrix)

ถ้า $A = \left[a_{ij}\right]_{m \times n}$ แล้ว ทรานสโพสของ A เขียนแทนด้วย $A^{'}, A^{t}$ หรือ A^{T} โดยที่ $A^{'} = \left[a_{ji}\right]_{n \times m}$ หรือ อธิบายง่ายๆคือการสลับสมาชิกแนวนอนเป็นแนวตั้งนั่นเอง

ตัวอย่าง 2.14 ให้ $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ จะได้ $A^{'} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

2.2 Matrix and python

เราสามารถสร้างเมทริกซ์ได้จาก numpy array หรือ numpy.matrix

คำสั่งเริ่มต้นที่ต้องสั่งคือ import numpy as np

2.2.1 คำสั่งและ function เกี่ยวกับการสร้าง matrix

<table>
<tr><th>function และ sourcecode</th><th>อธิบาย และ output</th></tr>
<tr><td>np.array
np.matrix<pre>a = np.array([[1,2,3], [4,5,6]])
print(a)
print(type(a))

a = np.matrix('1 2 3; 4 5 6')
print(a)
print(type(a))</pre></td><td>สร้าง matrix โดยกำหนดสมาชิก
ใน matrix<pre>[[1 2 3]
 [4 5 6]]
<class 'numpy.ndarray'>

[[1 2 3]
 [4 5 6]]
<class 'numpy.matrix'></pre></td></tr>
<tr><td>np.zeros<pre>a = np.zeros((2,3))
b = np.zeros((3,3))
print(a)
print(b)</pre></td><td>สร้าง zero matrix (สมาชิกเป็น 0
ทั้งหมด)<pre>[[0. 0. 0.]
 [0. 0. 0.]]

[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]</pre></td></tr>
<tr><td>np.ones<pre>a = np.ones((2,1))
b = np.ones((1,4))
print(a)
print(b)</pre></td><td>สร้าง one matrix (สมาชิกเป็น 1
ทั้งหมด)<pre>[[1.]
 [1.]]

[[1. 1. 1. 1.]]</pre></td></tr>
</table>

np.arange(x)
 	
np.reshape(x, y)
 	
`a = np.arange(5)`
`b = np.arange(12).reshape(3,4)`
`print(a)`
`print(b)`
	arange: สร้าง array ขนาด x และ initialize ค่าใน array เป็น 0 ถึง x-1
	reshape: แปลงขนาดของ array เป็น matrix ขนาด x แถว y คอลัมน์
	`[0 1 2 3 4]`
	
	`[[ 0  1  2  3]`
	` [ 4  5  6  7]`
	` [ 8  9 10 11]]`

np.shape
`A = np.arange(12).reshape(3,4)`
`print(np.shape(A))`
`print(A.shape[0])`
`print(A.shape[1])`
	แสดงขนาดของเมทริกซ์
	`(3, 4)`
	`3`
	`4`

np.identity(i) หรือ
np.eye(i)
`A = np.identity(3)`
`B = np.eye(2)`
`print(A)`
`print(B)`
	สร้าง identity matrix ขนาด i
	`[[1. 0. 0.]`
	` [0. 1. 0.]`
	` [0. 0. 1.]]`
	
	`[[1. 0.]`
	` [0. 1.]]`

np.random.rand(d0, d1, … dn)
 	
`A = np.random.rand(3,2)`
`B = np.random.rand(2,1)`
`print(A)`
`print(B)`
	สร้าง matrix โดย initialize ค่าแบบ random float
	
	`[[0.2667155  0.87270521]`
	` [0.5318026  0.04003233]`
	` [0.76498442 0.13770674]]`
	
	`[[0.10541614]`
	` [0.63546584]]`

np.random.randint(low, high=None, size = None)
 	
`A = np.random.randint(1,5, size =(3,2))`
`B = np.random.randint(6,10, size = (2,4))`
`print(A)`
`print(B)`
	สร้าง matrix โดย initialize ค่าแบบ random int
	`[[2 4]`
	` [3 3]`
	` [4 2]]`
	
	`[[9 6 9 7]`
	` [6 6 6 8]]`

2.2.2 function และตัวอย่างเกี่ยวกับ matrix operations

function และ source code	อธิบาย และ output

การบวกเมทริกซ์
`A =np.array([[1,2,0],[2,4,6]])`
`B =np.array([[2,-2,7],[1,3,5]])`
`C = A + B`
`print(C)`
	`[[ 3  0  7]`
	` [ 3  7 11]]`

การคูณเมทริกซ์
ใช้ .dot() หรือ np.matmul()
`A = np.array([[1,2,0],[2,4,6]])`
`B = np.array([[2,1],[1,7],[2,4]])`
`C = A.dot(B)`
`D = np.matmul(A,B)`
`print(C)`
`print(D)`
	`[[ 4 15]`
	` [20 54]]`
	
	`[[ 4 15]`
	` [20 54]]`

ทรานสโพสของเมทริกซ์
`A =np.array([[1,2,0],[2,4,6]])`
`B =np.array([[2,1], [1,7],[2,4]])`
`print(A.transpose())`
`print(B.transpose())`
	`[[1 2]`
	` [2 4]`
	` [0 6]]`
	
	`[[2 1 2]`
	` [1 7 4]]`

อินเวอร์สของเมทริกซ์ ใช้ np.linalg.inv
`A =np.array([[2,0],[5, -1]])`
 	
`A_inv = np.linalg.inv(A)`
`print(A)`
`print(A_inv)`
 	
`B =np.array([[2,3], [4,6]])`
`print(B)`
`B_inv = np.linalg.inv(B)`
`File "E:\python_by_prymania\`
	`[[ 2  0]`
	` [ 5 -1]]`
	
	`[[ 0.5  0. ]`
	` [ 2.5 -1. ]]`
	
	เมทริกซ์ B ไม่สามารถ
	หา inverse matrix ได้

ดีเทอร์มิแนนท์ของเมทริกซ์ ใช้ np.linalg.det

```python
A =np.array([[2,0],[5, -1]])

a = np.linalg.det(A)
print(A)
print(a)
```

```
[[ 2  0]
 [ 5 -1]]

-2.0
```

2.2.3 การเข้าถึงสมาชิกในเมทริกซ์ (matrix elements, rows and columns)

function และ sourcecode	อธิบาย และ output

การเข้าถึง element โดยใช้ [i] เลข index ของ element เริ่มนับที่ 0 ถ้า i เป็นลบ จะเริ่มนับที่ element สุดท้าย (-1 คือตัวสุดท้าย, -2 คือตัวรองสุดท้าย)

```python
A =np.array([[2,0,-1], [4,1,2], [-5,3,7]])

print(A,'\n')

#access elements of matrix
print('A[0][0] = ', A[0][0])
print('A[0][2] = ', A[0][2])
print('A[1][0] = ', A[1][0])
print('A[-1][-1] = ', A[-1][-1])
print('A[-1][-2] = ', A[-1][-2])
print('\n')
#access rows of matrix
print('A[0] = ', A[0])
print('A[1] = ', A[1])
print('A[2] = ', A[2])
print('A[-1] = ', A[-1])
print('\n')
#access columns of matrix
print('A[:,0] = ', A[:,0])
print('A[:,1] = ', A[:,1])
print('A[:,2] = ', A[:,2])
print('A[:,-1] = ', A[:,-1])
```

```
[[ 2  0 -1]
 [ 4  1  2]
 [-5  3  7]]

A[0][0] =  2
A[0][2] =  -1
A[1][0] =  4
A[-1][-1] =  7
A[-1][-2] =  3


A[0] =  [ 2  0 -1]
A[1] =  [4 1 2]
A[2] =  [-5  3  7]
A[-1] =  [-5  3  7]


A[:,0] =  [ 2  4 -5]
A[:,1] =  [0 1 3]
A[:,2] =  [-1  2  7]
A[:,-1] =  [-1  2  7]
```

การตัด matrix โดยการบอกจำนวน row, column

```python
A =np.array([[1,2,3,4], [5,6,7,8],
            [9,10,11,12],[13,14,15,16]])

print(A,'\n')

print(A[:1, :1])    # 1 row, 1 col
print(A[:1, :2])    # 1 row, 2 cols
print(A[:1, :3])    # 1 row, 3 cols
print(A[:1, :4])    # 1 row, 4 cols

print(A[:2, :1])    # 2 rows, 1 col
print(A[:2, :2])    # 2 rows, 2 cols
print(A[:2, :4])    # 2 rows, 4 cols

print(A[:3, :1])    # 3 rows, 1 col
print(A[:3, :2])    # 3 rows, 2 col

print(A[:, 0:2])    # all rows, col 0,1
print(A[:, 1:2])    # all rows, col 1
print(A[:, 1:4])    # all rows, col 1,2,3
print(A[:, 2:3])    # all rows, col 2
print(A[2:4, 2:4])  # row 2,3 , col 2,3
print(A[2:9, 2:8])  # row 2,3 , col 2,3
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]

[[1]]
[[1 2]]
[[1 2 3]]
[[1 2 3 4]]

[[1]
 [5]]

[[1 2]
 [5 6]]

[[1 2 3 4]
 [5 6 7 8]]

[[1]
 [5]
 [9]]

[[ 1  2]
 [ 5  6]
 [ 9 10]]

[[ 1  2]
 [ 5  6]
 [ 9 10]
 [13 14]]

[[ 2]
 [ 6]
 [10]
 [14]]

[[ 2  3  4]
 [ 6  7  8]
 [10 11 12]
 [14 15 16]]
[[ 3]
 [ 7]
 [11]
 [15]]

[[11 12]
 [15 16]]

[[11 12]
 [15 16]]
```

2.2.4 ฟังชันอื่น ๆ ที่สำคัญเกี่ยวกับ matrix

function และ sourcecode	อธิบาย และ output

argmax() แสดง index ที่มีค่ามากที่สุดใน matrix
amax() แสดงค่าที่มากที่สุดใน matrix
argmin() แสดง index ที่มีค่าน้อยที่สุดใน matrix
amin() แสดงค่าที่น้อยที่สุดใน matrix
 	
`A = np.random.randint(1,20, size = (3,5))`
`i_max = np.argmax(A)`
`v_max = np.amax(A)`
`i_min = np.argmin(A)`
`v_min = np.amin(A)`
 	
`print(A)`
`print(i_max)`
`print(v_max)`
`print(i_min)`
`print(v_min)`
	`[[14 13  8  6 18]`
	` [18  7 17 17 16]`
	` [17 16  1  5 16]]`
	
	`4`
	`18`
	
	`12`
	`1`
	
	เมทริกซ์ A ค่าที่มากที่สุดคือ 18 ซึ่งอยู่ในตำแหน่ง index ที่ 4 ดังนั้น
	argmax คือ 4
	amax คือ 18

sum(axis = 0) หาผลบวกรวมในเมทริกซ์ แต่ละ column
sum(axis = 1) หาผลบวกรวมในเมทริกซ์ แต่ละ row
`A = np.array([[1,2,3,4],[5,6,7,8]])`
`s1 = A.sum(axis = 0)`
`s2 = A.sum(axis = 1)`
`print(A)`
`print(s1)`
`print(s2)`
	`[[1 2 3 4]`
	` [5 6 7 8]]`
	
	`[ 6  8 10 12]`
	
	`[10 26]`

2.3 แบบฝึกหัด

1. หาค่าของ $A + B$ และ $A \times B$ (ถ้าสามารถหาได้)

Q	A, B	$A + B$	$A \times B$
1.1	$A = \begin{bmatrix} 1 & 2 \\ -1 & -2 \end{bmatrix}, B = \begin{bmatrix} 0 & -2 \\ -1 & -2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ -2 & -4 \end{bmatrix}$	$\begin{bmatrix} -2 & -6 \\ 2 & 6 \end{bmatrix}$
1.2	$A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & -2 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & -2 & 0 \\ -1 & -2 & 2 \end{bmatrix}$		
1.3	$A = \begin{bmatrix} 1 & 2 \\ -1 & -2 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 & -2 & 0 \\ -1 & -2 & 7 \end{bmatrix}$		
1.4	$A = \begin{bmatrix} 1 & -1 \\ 2 & 4 \end{bmatrix}, B = \begin{bmatrix} -1 & 0 & 2 \\ 2 & 4 & 6 \end{bmatrix}$		

2. หาค่าของ $\det(A)$, A^{-1} และ A^{t} (ถ้าสามารถหาได้)

Q	A	$\det(A)$	A^{-1}	A^{-t}
2.1	$\begin{bmatrix} 1 & 2 \\ -1 & -2 \end{bmatrix}$	0		$\begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix}$
2.2	$\begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$	-6		
2.3	$\begin{bmatrix} 1 & 2 & 4 \\ 2 & -2 & 0 \\ 1 & -2 & -1 \end{bmatrix}$			

3. จากเมทริกซ์ที่กำหนดให้ ให้ค่าของตัวแปร a และ b

Q	เมทริกซ์	คำตอบ
3.1	$\begin{bmatrix} b+2 & 1 \\ -1 & b-1 \end{bmatrix} = \begin{bmatrix} a-5 & a+b \\ a-5 & -4 \end{bmatrix}$	a = b =
3.2	$\begin{bmatrix} 2 & 5 \\ -4 & -3 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -1 \\ -5 \end{bmatrix}$	a = b =

4. เขียนโปรแกรมด้วย python เพื่อทำงานต่อไปนี้

Q	การทำงาน
4.1	สร้างเมทริกซ์ A ขนาด 3 rows, 4 columns โดย initialize ค่า (กำหนดค่าเริ่มต้น) เป็นเท่าไรก็ได้ สร้างเมทริกซ์ B ขนาด 4 rows, 2 columns โดย initialize ค่าเป็นเท่าไรก็ได้ C = A × B แสดง output เป็นเมทริกซ์ A, B และ C **<u>ตัวอย่าง output</u>** A [[1 2 3 4] [-1 -2 -3 0] [2 3 6 7]] B [[0 -2] [-7 -6] [3 6] [0 5]] C = A x B [[-5 24] [5 -4] [-3 49]]
4.2	ใช้ฟังก์ชัน arange() และ reshape() เพื่อ สร้างเมทริกซ์ A ขนาด 3 rows, 2 columns สร้างเมทริกซ์ B ขนาด 3 rows, 2 columns C = A + B แสดง output เป็นเมทริกซ์ A, B และ C **<u>ตัวอย่าง output</u>**

	A [[0 1] [2 3] [4 5]] B [[0 1] [2 3] [4 5]] C = A + B [[0 2] [4 6] [8 10]]
4.3	ใช้ฟังก์ชัน np.random.randint() และกำหนดค่า parameters ที่เหมาะสม เพื่อ สร้างเมทริกซ์ A ขนาด 3 rows, 4 columns โดยสมาชิกในเมทริกซ์มีค่าอยู่ระหว่าง 1 ถึง 10 สร้างเมทริกซ์ B ขนาด 3 rows, 4 columns โดยสมาชิกในเมทริกซ์มีค่าอยู่ระหว่าง 2 ถึง 5 C = A + B D = A – B แสดง output เป็นเมทริกซ์ A, B, C และ D **ตัวอย่าง output** A [[5 4 6 1] [2 6 6 5] [7 2 8 6]] B [[2 4 2 4] [4 4 2 4] [3 2 3 2]] C = A + B [[7 8 8 5] [6 10 8 9] [10 4 11 8]] D = A - B [[3 0 4 -3] [-2 2 4 1] [4 0 5 4]]

4.4	ใช้ฟังก์ชัน np.random.randint() และกำหนดค่า parameters ที่เหมาะสม เพื่อ สร้างเมทริกซ์ A ขนาด 3 rows, 3 columns โดยสมาชิกในเมทริกซ์มีค่าอยู่ระหว่าง -5 ถึง 5 สร้างเมทริกซ์ B ขนาด 3 rows, 2 columns โดยสมาชิกในเมทริกซ์มีค่าอยู่ระหว่าง -2 ถึง 2 แสดง output ต่อไปนี้ เมทริกซ์ A เมทริกซ์ B A × B transpose ของเมทริกซ์ A inverse ของเมทริกซ์ A determinant ของเมทริกซ์ A **ตัวอย่าง output**

```
A
[[ 2 -1  3]
 [ 3 -1  0]
 [-2  2 -2]]

B
[[ 1 -1]
 [-1 -1]
 [-2  1]]

A x B
[[-3  2]
 [ 4 -2]
 [ 0 -2]]

Transpose of A
[[ 2  3 -2]
 [-1 -1  2]
 [ 3  0 -2]]

Inverse of A
[[ 0.2  0.4  0.3]
 [ 0.6  0.2  0.9]
 [ 0.4 -0.2  0.1]]

Determinant of A
10.000000000000002
```

4.5	สร้างเมทริกซ์ A ขนาด 3 × 4 โดยกำหนดค่าในเมทริกซ์ดังนี้ $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 8 & 7 & 6 & 5 \\ -2 & -4 & -6 & -8 \end{bmatrix}$ แสดง output ของเมทริกซ์ A ต่อไปนี้ โดยใช้ฟังก์ชันที่เหมาะสม - เมทริกซ์ A - row ที่ index 0 - row สุดท้าย (โดยไม่ให้บอกว่าเป็น index ที่ 2) - column ที่ index 1 - column สุดท้าย (โดยไม่ให้บอกว่าเป็น index ที่ 3) - เลข 7 (โดยการบอก row, column ของเมทริกซ์) - สมาชิกตัวสุดท้ายของเมทริกซ์ (เลข -4) - row ที่ index 0 และ 1 - column ที่ index 1 และ 2 - index ที่มีค่ามากที่สุดในเมทริกซ์ A - ค่าที่มากที่สุดในเมทริกซ์ A - index ที่มีค่าน้อยที่สุดในเมทริกซ์ A - ค่าที่น้อยที่สุดในเมทริกซ์ A - ผลรวมของเมทริกซ์แต่ละ column - ผลรวมของเมทริกซ์แต่ละ row **ตัวอย่าง output**

```
A
[[ 1  2  3  4]
 [ 8  7  6  5]
 [-2 -4 -6 -8]]

row#0:     [1 2 3 4]

last row:  [-2 -4 -6 -8]

column#1:  [ 2  7 -4]

last col:  [ 4  5 -8]

A[1][2]:   6

last element:  -8

row #0,1:
 [[1 2 3 4]
 [8 7 6 5]]

col #1,2:
 [[ 2  3]
 [ 7  6]
 [-4 -6]]

index of max value:  4

max value:           8

index of min value:  11

min value:           -8

sum of each column: [7 5 3 1]

sum of each row: [ 10  26 -20]
```

2.4 References

https://www.mathsisfun.com/algebra/matrix-inverse.html

https://www.shelovesmath.com/algebra/advanced-algebra/matrices-and-solving-systems-with-matrices/#AddingandSubtractingMatrices

https://www.intmath.com/matrices-determinants/5-inverse-matrix.php

https://th.wikihow.com/%E0%B8%AB%E0%B8%B2%E0%B8%94%E0%B8%B5%E0%B9%80%E0%B8%97%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%A1%E0%B8%B4%E0%B9%81%E0%B8%99%E0%B8%99%E0%B8%95%E0%B9%8C%E0%B8%82%E0%B8%AD%E0%B8%87%E0%B9%80%E0%B8%A1%E0%B8%97%E0%B8%A3%E0%B8%B4%E0%B8%81%E0%B8%8B%E0%B9%8C%E0%B8%A1%E0%B8%B4%E0%B8%95%E0%B8%B4-3x3