

Matrix Multiplication

เมทริกซ์ของจำนวนจริงขนาด $p \times q$ สามารถแทนได้ด้วยลิสต์ขนาด p ช่อง โดยที่แต่ละช่องเก็บลิสต์ขนาด q ช่อง เช่น

$$\begin{bmatrix} 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 4 & 1 & 2 & 2 \end{bmatrix} \quad \text{แทนได้ด้วย} \quad \begin{bmatrix} [1, 2, 3, 0], \\ [2, 3, 0, 1], \\ [4, 1, 2, 2] \end{bmatrix}$$

จงเขียนฟังก์ชัน

- `mult_c(c, A)` ที่คืนเมทริกซ์ที่เป็นผลจากการคูณจำนวนจริง `c` กับเมทริกซ์ `A`
- `mult(A, B)` ที่คืนเมทริกซ์ที่เป็นผลจากการคูณเมทริกซ์ `A` กับ `B` (ซึ่งคือ $A \times B$)

```
def read_matrix():
    m = []
    n_rows = int(input())
    for k in range(n_rows):
        x = input().split()
        r = []
        for e in x:
            r.append(float(e))
        m.append(r)
    return m

def mult_c(c, A):

def mult(A, B):

exec(input().strip()) # ต้องมีคำสั่งนี้ ตรงนี้ ดอนส่งให้ Grader ตรวจ
```

From wikipedia (https://en.wikipedia.org/wiki/Matrix_multiplication)

Definition [edit]

If **A** is an $n \times m$ matrix and **B** is an $m \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mp} \end{pmatrix}$$

the *matrix product* $\mathbf{C} = \mathbf{AB}$ (denoted without multiplication signs or dots) is defined to be the $n \times p$ matrix

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{pmatrix}$$

such that

$$c_{ij} = a_{i1}b_{1j} + \cdots + a_{im}b_{mj} = \sum_{k=1}^m a_{ik}b_{kj},$$

for $i = 1, \dots, n$ and $j = 1, \dots, p$.

ข้อมูลนำเข้า

คำสั่งภาษา Python ที่ใช้ทดสอบการทำงานของฟังก์ชัน

ข้อมูลส่งออก

ผลที่ได้จากการสั่งทำงานคำสั่งที่ได้รับ

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
A=read_matrix() ;print(mult_c(0.5,A)) 3 1 2 2 3 3 2	[[0.5, 1.0], [1.0, 1.5], [1.5, 1.0]]
A=read_matrix() ;B=read_matrix() ;print(mult(A,B)) 3 1 2 3 1 1 1 2 2 2 3 1 2 2 3 3 2	[[14.0, 14.0], [6.0, 7.0], [12.0, 14.0]]