

Q4_Sample_P4_Numpy

ฟังก์ชัน `read_matrix()` ข้างล่างนี้ อ่านข้อมูลของ **matrix** มาสร้าง **numpy array** แบบ 2 มิติ ดังตัวอย่างข้างล่างนี้

input	ผลที่ได้จาก <code>read_matrix()</code>
<pre> 4 6 2 1 0 3 1 7 1 4 2 1 0 3 3 3 2 5 4 0 0 2 3 2 4 1 </pre>	<pre> array([[2, 1, 0, 3, 1, 7], [1, 4, 2, 1, 0, 3], [3, 3, 2, 5, 4, 0], [0, 2, 3, 2, 4, 1]]) </pre>

จงเขียนฟังก์ชันต่าง ๆ ที่เว้นว่างในโปรแกรมข้างล่างนี้ ที่มีข้อกำหนดของพารามิเตอร์ และผลลัพธ์ที่ได้ตามตารางนี้

function	input parameter	return value
<code>dot_row_column(m)</code>	m คือ square matrix	numpy array p ขนาดเดียวกันกับ m โดย <code>p[i][j]</code> มีค่าเท่ากับการนำ m แถวที่ i dot กับ m คอลัมน์ที่ j
<code>mul_submatrix(m,n)</code>	m คือ matrix ขนาด r x c n คือ square matrix ขนาด k x k โดยที่ r และ c ทหารด้วย k ลงตัว	numpy array p ขนาดเดียวกันกับ m โดยนำ n คูณแบบเมทริกซ์กับเมทริกซ์ย่อยขนาด k x k ของ m เช่น ถ้า m คือเมทริกซ์ตัวอย่างด้านบน และ n คือ <code>array([[1,2],[0,1]])</code> จะได้ <div style="display: flex; flex-direction: column; align-items: center;"> <div> $\begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 5 \\ 1 & 6 \end{bmatrix}$ </div> <div> $\begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ 2 & 5 \end{bmatrix}$ </div> <div> $\begin{bmatrix} 1 & 7 \\ 0 & 3 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 9 \\ 0 & 3 \end{bmatrix}$ </div> <div> $\begin{bmatrix} 3 & 3 \\ 0 & 2 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 9 \\ 0 & 2 \end{bmatrix}$ </div> <div> $\begin{bmatrix} 2 & 5 \\ 3 & 2 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 9 \\ 3 & 8 \end{bmatrix}$ </div> <div> $\begin{bmatrix} 4 & 0 \\ 4 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 4 & 9 \end{bmatrix}$ </div> </div> <p>และได้ p คือ</p> <pre> array([[2, 5, 0, 3, 1, 9], [1, 6, 2, 5, 0, 3], [3, 9, 2, 9, 4, 8], [0, 2, 3, 8, 4, 9]]) </pre>
<code>resize(m,a,b)</code>	m คือ matrix ใด ๆ a, b คือ จำนวนเต็มบวก	numpy array p ขนาด a x b ซึ่งเกิดจากการขยาย/ตัด m โดยหากเป็นการขยาย ให้ทำการซ้ำแถว/คอลัมน์จนได้ขนาดที่กำหนด

```

import numpy as np

def read_matrix():
    r,c = [int(e) for e in input().split()]
    m = [[int(e) for e in input().split()] for i in range(r)]
    return np.array(m)

def dot_row_column(m):
    # write your code here

def mul_submatrix(m,n):
    # write your code here

def resize(m,a,b):
    # write your code here

exec(input().strip()) # do not remove this line

```

ข้อมูลนำเข้า

คำสั่งในการทดสอบฟังก์ชันที่เขียน

ข้อมูลส่งออก

ผลที่ได้จากคำสั่งที่ป้อนเป็นข้อมูลนำเข้า

ตัวอย่าง

input	output
<pre> m=read_matrix();print(dot_row_column(m)) 3 3 1 0 6 3 -1 4 2 5 2 </pre>	<pre> [[13 30 18] [8 21 22] [21 5 36]] </pre>
<pre> m=read_matrix();n=read_matrix();print(mul_submatrix(m,n)) 4 6 2 1 0 3 1 7 1 4 2 1 0 3 3 3 2 5 4 0 0 2 3 2 4 1 2 2 1 2 0 1 </pre>	<pre> [[2 5 0 3 1 9] [1 6 2 5 0 3] [3 9 2 9 4 8] [0 2 3 8 4 9]] </pre>
<pre> m=read_matrix();print(resize(m,2,4)) 4 6 2 1 0 3 1 7 1 4 2 1 0 3 3 3 2 5 4 0 0 2 3 2 4 1 </pre>	<pre> [[2 1 0 3] [1 4 2 1]] </pre>
<pre> m=read_matrix();print(resize(m,6,9)) 4 6 2 1 0 3 1 7 1 4 2 1 0 3 3 3 2 5 4 0 0 2 3 2 4 1 </pre>	<pre> [[2 1 0 3 1 7 2 1 0] [1 4 2 1 0 3 1 4 2] [3 3 2 5 4 0 3 3 2] [0 2 3 2 4 1 0 2 3] [2 1 0 3 1 7 2 1 0] [1 4 2 1 0 3 1 4 2]] </pre>