Assignment 3 - 2D-Array

IN PROGRESS
0% ATTEMPT RATE
0/500 GRADE

Operation for Matrix

The concept of matrices is essential in linear algebra and computer science.

Background

All matrix operations are derived from vector operations. In this part, a **one-dimensional array** is used to represent a vector. The addition, subtraction, and multiplication operations are described as follows.

Addition:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_m + b_m \end{bmatrix}$$

Subtraction:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} a_1 - b_1 \\ a_2 - b_2 \\ \vdots \\ a_m - b_m \end{bmatrix}$$

Multiplication:

$$a * b = \begin{bmatrix} a_1 & a_2 & \cdots & a_m \end{bmatrix} * \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = a_1 * b_1 + a_2 * b_2 \cdots + a_m * b_m$$

Then, let's expand vectors into a **two-dimensional matrix**. In this case, the matrices are stored by using two-dimensional arrays (only square matrices are considered, that is, $A \in R^{nxn}$). The addition and subtraction operations are the same as the corresponding vector ones. Furthermore, two new operations, **transpose** and **multiplication**, are introduced.

Transpose:

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

Multiplication:

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, a_k \in R^n$$

$$A * B^T = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} * \begin{bmatrix} b_1^T & b_2^T & \cdots & b_n^T \end{bmatrix} = \begin{bmatrix} a_1 * b_1^T & \cdots & \cdots & a_1 * b_n^T \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ a_n * b_1^T & \cdots & \cdots & a_n * b_n^T \end{bmatrix}$$

Description

In this problem, we define a brand new new operator `#`. The following equation shows how `#` works on two matrices A and B:

$$A\#B = \begin{bmatrix} C_1 & C_2 \\ C_3 & C_4 \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \# \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} = \begin{bmatrix} A_1 + B_1 & A_2 * B_2^T \\ B_3 * A_3^T & A_4 - B_4 \end{bmatrix}$$

> **Example**
>
> $$\begin{bmatrix} 22 & 3 & 89 & 5 \\ 5 & 20 & 18 & 6 \\ 5 & 17 & 222 & 10 \\ 1 & 6 & 0 & 334 \end{bmatrix} \# \begin{bmatrix} 4 & 2 & 12 & 50 \\ 53 & 0 & 432 & 70 \\ 67 & 9 & 43 & 100 \\ 213 & 92 & 876 & 1 \end{bmatrix} = \begin{bmatrix} 26 & 5 & 1318 & 38798 \\ 58 & 20 & 516 & 8196 \\ 488 & 121 & 179 & -90 \\ 2629 & 765 & -876 & 333 \end{bmatrix}$$

The task of this problem is to compute the result of `#` operations on A and B. We guarantee that $A, B \in R^{2nx2n}$, $A_i, B_i \in R^{nxn}$, try to make the corresponding operation

Input

- The first row gives `n`, the size of the two matrices.
- The following `n` line are the elements of the first matrix (noted as A).
- The following `n` line are the elements of the second matrix (noted as B).
- All elements in all matrices are integers.

For all test cases, We guarantee that

1. n is a multiple of 2
2. $0 < n \leq 100$
3. $0 \leq a_{ij} \leq 100$
4. $0 \leq b_{ij} \leq 100$

Output

Output the result of $A\#B$.

Sample

Input #1 Copy

```
4
0 4 10 5
8 5 7 6
8 5 7 4
0 3 9 5
9 5 4 2
0 8 1 2
1 6 4 2
3 3 7 8
```

Output #1 Copy

```
9 9 50 20
8 13 40 19
38 18 3 2
39 9 2 -3
```

Input #2 Copy

```
2
5 3
7 2
7 0
4 0
```

Output #2 Copy

```
12 0
28 2
```


Problem Information

Problem Type	Java Only
Time Limit	2000 ms
Memory Limit	128 MB
IO Mode	Stream IO
Difficulty	Unknown
Sources	