Exercise P5.2 *Submatrix Sum.*

You are given a $n \times n$ matrix $M = (m_{i,j})$ in which each entry $m_{i,j}$ with $1 \leq i, j \leq n$ is an integer between 0 and 1000 (rows and columns are numbered from 1 to n, from top-left to bottom-right). Your task is to design a data structure that, after *preprocessing* the matrix M, is able to support the following *query* operation: *Given* $a, b, c, d \in \mathbb{Z}$ *with* $1 \leq a \leq b \leq n$ *and* $1 \leq c \leq d \leq n$, *return*

$$S(a, b, c, d) = \sum_{\substack{a \leq i \leq b \\ c \leq j \leq d}} m_{i,j}.$$

Input The first line of the input contains the integer n. Each of the following n lines is one row of M. More precisely, the $(i+1)$-th line of the input contains the n integers $m_{i,1}, \ldots, m_{i,n}$. The $(n+2)$-th line of the input contains the number m of queries to be answered and the i-th of the following m lines ($1 \leq i \leq m$) contains four integers a_i, b_i, c_i, d_i.

Output

The output consists of m lines, where the i-th line contains the answer to the i-th query, i.e., the number $S(a_i, b_i, c_i, d_i)$.

Grading You get 2 bonus points if your program works for all inputs. Your algorithm should require time $O(n^2)$ preprocessing time and it should answer each query in constant time. Submit your `Main.java` at `https://judge.inf.ethz.ch/team/websubmit.php?cid=18997&problem=DA17P4.2`. The enrollment password is "`asymptotic`".

Example

5	3	1	5	0
8	0	4	3	6
1	6	1	5	1
0	7	9	1	7
4	5	8	8	3

Table 1: Example of matrix M with $n = 5$.

Input (corresponding to the matrix in Table 1):

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

Output:

```
59
38
4
```

Notes For this exercise we provide an archive on the lecture website containing a program template that will load the input and write the output for you. The archive also contains additional test cases (which differ from the ones used for grading). Importing any additional Java class is **not allowed** (with the exception of the already imported `java.util.Scanner` class).