7716 The Cure

It has been 4 years since the first alien abduction. Last year we discovered that some of the newborn babies whose parents have been abducted, possess alien DNA molecules! The Earth government managed to quarantine all the alien babies in a secret place. With the latest advancement in bio engineering, we now have a way to cure the alien babies and safely return them back to their parents.

As usual, the Earth government put the brilliant scientist in charge to lead the engineering of the cure. There are so many challenging problems to engineer a perfect cure. The cure must be personalized for each baby by taking account to their current biological state. But, don't worry! The brilliant scientist has figured out all the solutions... except for one critical problem:

The brilliant scientist has a device that scans a region of the baby's cells and produces an $N \times N$ matrix of integers. To cure this cell region, a quick analysis must be done by running Q queries where each query asks for the k-th smallest element inside a subregion of the $N \times N$ matrix. The analysis result is then used to apply the appropriate cure for the cell region. The problem is that the number of queries Q may be very large. Since the cure is time sensitive, if the analysis takes too long, it became worthless because the current state of the cell may have changed too much by then.

The brilliant scientist discovered the problem above when testing the device to cure the first baby shown on the right, named Charliezt. The brilliant scientist is well aware on the importance of programming for several years now, but never got the time to learn it yet. The brilliant scientist is reluctant to ask for your help since last year contestants' performance were disappointing. However, the secretary forwarded this problem to you anyway due to its importance. This is your last chance to gain the brilliant scientist trust back!

Input

The input contains only a single case with multiple queries. The first line is an integer N ($1 \leq N \leq 250$).

The next N lines, each contains N integers between 1 and 10,000, inclusive, separated by a space representing the values of the $N \times N$ matrix. The next line is an integer Q ($1 \leq Q \leq 250,000$) denoting the number of queries. The next Q lines, each represents a query. Each query analyzes a square region inside the matrix represented by 4 integers: r, c, s, k. Where:

- r and c is the top left coordinate of the square. The coordinate is 1-based. For example, coordinate $(3, 6)$ means the top left position of the square is at the 3rd row from the top and 6th column from the left,
- s is the size of the square,
- k ($1 \leq k \leq s^2$) is the k-th smallest element you need to find inside the square.

You may assume all queries represent valid square sub-matrices of the given matrix.

Output

For each query output one line containing the value of the k-th smallest element in the given square in the query.

Explanation for the sample:

First query: 1 3 2 4. This query asks for the 4th smallest element in a square matrix of size 2 where the topleft coordinate is at (1, 3). The square matrix being queried is:

```
23 8
17 27
```

and the 4th smallest element in the square matrix query is 27.

Second query: 1 1 5 14. This query asks for the 14^{th} smallest element in a square matrix of size 5 where the top-left coordinate is at (1, 1). The square matrix being queried is:

```
25 13 23  8 39
29 11 17 27 26
24 33  5 14 11
11  1 23 21  6
38 38 40  4 16
```

and the 14th smallest element in the square matrix query is 23.

Sample Input

```
5
25 13 23 8 39
29 11 17 27 26
24 33 5 14 11
11 1 23 21 6
38 38 40 4 16
10
1 3 2 4
1 1 5 14
2 1 4 12
3 4 1 1
2 2 2 1
4 2 2 2
2 1 3 8
1 2 4 12
2 4 1 1
2 1 4 15
```

Sample Output

```
27
23
29
14
5
23
29
23
27
38
```