

You are given two integers 'N' and 'K', also provided with a 'N x N' square matrix 'ARR'.

Your task is to print the sum of all sub-squares of size 'K x K' where 'K' is smaller than or equal to 'N'.

Input Format:

The first line contains an integer 'T', which denotes the number of test cases to be run. Then, the 'T' test cases follow.

The first line of each test case contains two positive integers, 'N' and 'K' denoting the number of rows in the square matrix 'ARR' and the number of rows in the square sub-matrix respectively.

For each test case, the next 'N' lines contain N numbers each denoting the elements of the matrix 'ARR'.

Output Format:

For each test case return a 2D array that stores 'N - K' arrays each containing 'N - K' elements, where the number in the ith array at the jth position denotes the sum of the sub-matrix whose left corner is at index [i, j].

The output of each test case will be printed in a separate line.

Constraints:

```
1 <= 'T' <= 10
1 <= 'N' <= 500
1 <= 'K' <= 'N'
1 <= 'ARR[i][j]' <= 1000
```

Where 'ARR[i][j]' denotes the matrix element at the jth column in the ith row

Time Limit: 1 sec

Sample Input 1:

```
1
3 2
8 1 3
2 9 3
0 3 5
```

Sample Output 1:

```
20 16
14 20
```

Explanation For Sample Input 1:

There can be a total of 4 sub-matrix of size 2x2:

First, starting at index [0,0]:

```
8 1
```

```
2 9
```

So the sum of this matrix will be $(8+1+2+9) = 20$.

Second, starting at index [0,1]:

```
1 3
```

```
9 3
```

So the sum of this matrix will be $(1+3+9+3) = 16$.

Third, starting at index [1,0]:

```
2 9
```

```
0 3
```

So the sum of this matrix will be $(2+9+0+3) = 14$

Fourth, starting at index [1,1]:

```
9 3
```

```
3 5
```

So the sum of this matrix will be $(9+3+3+5) = 20$.

So we will return 2D array of size 2*2 with values as calculated above.

Sample Input 2:

```
1
2 2
5 7
8 1
```

Sample Output 2:

```
21
```

Explanation For Sample Input 2:

Only 1 sub-matrix is possible starting from index [0, 0] sum of which is 21.