

## Problem

Arsh recently found an old rectangular circuit board that he would like to recycle. The circuit board has  $R$  rows and columns of squares.

Each square of the circuit board has a thickness, measured in millimetres. The square in the  $r$ -th row and  $c$ -th column has thickness  $V_{r,c}$ . A circuit board is good if in each row, the difference between the thickest square and the least thick square is no greater than  $K$ .

Since the original circuit board might not be good, Arsh would like to find a good subcircuit board. A subcircuit board can be obtained by choosing an axis-aligned subrectangle from the original board and taking the squares in that subrectangle. Arsh would like your help in finding the number of squares in the largest good subrectangle of his original board.

## Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow. Each test case begins with one line containing three integers  $R$ ,  $C$  and  $K$ , the number of rows, the number of columns, and the maximum difference in thickness allowed in each row.

Then, there are  $R$  more lines containing  $C$  integers each. The  $c$ -th integer on the  $r$ -th line is  $V_{r,c}$ , the thickness of the square in the  $r$ -th row and  $c$ -th column.

## Output

For each test case, output one line containing Case # $x$ :  $y$ , where  $x$  is the test case number (starting from 1) and  $y$  is the maximum number of squares in a good subrectangle.

## Limits

Time limit: 15 seconds per test set.

Memory limit: 1GB.

$1 \leq T \leq 50$ .

$1 \leq R \leq 300$ .

$1 \leq C \leq 300$ .

$0 \leq V_{i,j} \leq 103$  for all  $i, j$ .

Test set 1 (Visible)

$K = 0$ .

Test set 2 (Hidden)

$0 \leq K \leq 103$ .

## Sample

Input 1

Output 1

3

1 4 0

3 1 3 3

2 3 0

4 4 5

7 6 6

4 5 0

2 2 4 4 20  
8 3 3 3 12  
6 6 3 3 3  
1 6 8 6 4

Case #1: 2  
Case #2: 2  
Case #3: 6

Input 2

Output 2

3  
1 4 2  
3 1 3 3  
3 3 2  
0 5 0  
8 12 3  
7 10 1  
4 4 8  
20 10 20 10  
10 4 5 20  
20 5 4 10  
10 20 10 20

Case #1: 4  
Case #2: 3  
Case #3: 4

Note: Sample 2 is not valid for Test set 1. Sample 1 will be tested prior to running Test set 1 (the same way samples normally are). However, Sample 2 will not be tested prior to running Test set 2.

The sample cases are illustrated below. For each case, the good subcircuit board with the largest number of squares is highlighted in green.

```
#include <string>
#include <iostream>
#include <algorithm>
using namespace std;
const int inf = 1012345678;
int H, W, K, A[309][309]; bool ok[309][309][309];
int main() {
    cin.tie(0);
    ios_base::sync_with_stdio(false);
    int Q;
    cin >> Q;
    for (int rep = 1; rep <= Q; ++rep) {
```

```

cin >> H >> W >> K;
for (int i = 0; i < H; ++i) {
    for (int j = 0; j < W; ++j) {
        cin >> A[i][j];
    }
}
for (int i = 0; i < H; ++i) {
    for (int j = 0; j < W; ++j) {
        int cl = inf, cr = -inf;
        for (int k = j; k < W; ++k) {
            cl = min(cl, A[i][k]);
            cr = max(cr, A[i][k]);
            if (cr - cl <= K) {
                ok[i][j][k] = true;
            }
            else {
                ok[i][j][k] = false;
            }
        }
    }
}
int ans = 0;
for (int i = 0; i < W; ++i) {
    for (int j = i; j < W; ++j) {
        int cont = 0;
        for (int k = 0; k < H; ++k) {
            if (ok[k][i][j]) ++cont;
            else cont = 0;
            ans = max(ans, cont * (j - i + 1));
        }
    }
}
cout << "Case #" << rep << ": " << ans << endl;
}
return 0;
}

```