**Find A Specific Pair In Matrix**

**Difficulty: MEDIUM**

A picture containing text, clipart

Description automatically generated

Contributed By

**Avg. time to solve**

**10 min**

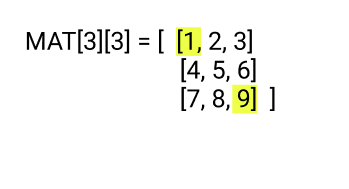
**Success Rate**

**90%**

**Problem Statement**

**You have been given a 2-D matrix ‘MAT’ of size ‘N’ x ‘N’ i.e. N rows and N columns. Your task is to find the maximum value of ‘MAT[a][b]’ - ‘MAT[c][d]’ over all possible indices (0 <= ‘a’, ‘b’, ‘c’, ‘d’ < ‘N’) such that . ‘a’ > ‘c’ and ‘b’ > ‘d’.**

**For example:**



In this example, to maximise the value of ‘MAT[a][b]’ - ‘MAT[c][d]’ fulfilling the given conditions on indices (‘a’ > ‘c’ and ‘b’ > ‘d’), we take ‘a’ = 2, ‘b’ = 2, ‘c’ = 0 and ‘d’ = 0 . So, ‘MAT[a][b]’ = 9 and ‘MAT[c][d]’ = 1 and the value of ‘MAT[a][b]’ - ‘MAT[c][d]’ => 9 - 1 = 8 which is maximum among all possible combinations.

**Input Format:**

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

The first line of each test case contains one integer ‘N’ representing the number of rows (or columns) of the square matrix ‘MAT’.

The next ‘N’ lines of each test case contain ‘N’ single space-separated integers denoting the values of ‘MAT’.

**Output Format :**

For each test case, print the maximum value of ‘MAT[a][b]’ - ‘MAT[c][d]’ over all possible choices of indices i.e. 0 <= ‘a’, ‘b’, ‘c’, ‘d’ < ‘N’ such that ‘a’ > ‘c’ and ‘b’ > ‘d’.

Print the output of each test case in a separate line.

**Note:**

You do not need to print anything; it has already been taken care of. Just implement the given function.

**Constraints:**

1 <= ‘T’ <= 100

2 <= ‘N’ <= 100

-10^5 <= ‘MAT[i][j]’ <= 10^5

Time Limit : 1 second

**Sample Input 1:**

2

3

1 2 3

4 5 6

7 8 9

3

-1 -2 -3

-4 -5 -6

-7 -8 -9

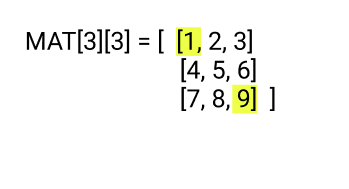
**Sample Output 1:**

8

-4

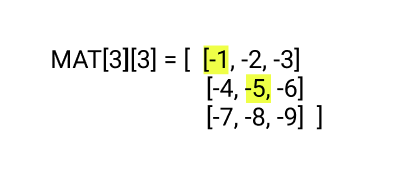
**Explanation For Sample Output 1:**

For sample test case 1:



In this sample test case, to maximise the value of ‘MAT[a][b]’ - ‘MAT[c][d]’ fulfilling the given conditions on indices (‘a’ > ‘c’ and ‘b’ > ‘d’), we take ‘a’ = 3, ‘b’ = 3, ‘c’ = 0 and ‘d’ = 0 . So, ‘MAT[a][b]’ = 9 and ‘MAT[c][d]’ = 1 and the value of ‘MAT[a][b]’ - ‘MAT[c][d]’ => 9 - 1 = 8 which is maximum among all possible combinations.

For sample test case 2:



In this sample test case, to maximise the value of ‘MAT[a][b]’ - ‘MAT[c][d]’ fulfilling the given conditions on indices (‘a’ > ‘c’ and ‘b’ > ‘d’), we take ‘a’ = 1, ‘b’ = 1, ‘c’ = 0 and ‘d’ = 0 . So, ‘MAT[a][b]’ = -5 and ‘MAT[c][d]’ = -1 and the value of ‘MAT[a][b]’ - ‘MAT[c][d]’ => (-5) - (-1) = -4 which is maximum among all possible combinations.

**Sample Input 2:**

2

2

1 5

4 2

3

-1 5 -3

-14 -5 -2

-7 8 -9

**Sample Output 2:**

1

22

**Previous**

int findMaxValue(vector<vector<int>>& mat, int n) {

    // Write your code here.

    vector<vector<int>> aux(n, vector<int> (n));

    int maxm=-100001;

    for (int i=n-1; i>=0; i--) {

        if (mat[n-1][i]>maxm) maxm=mat[n-1][i];

        aux[n-1][i]=maxm;

    }

    maxm=-100001;

    for (int i=n-1; i>=0; i--) {

        if (mat[i][n-1]>maxm) maxm=mat[i][n-1];

        aux[i][n-1]=maxm;

    }

    for (int i=n-2; i>=0; i--) {

        for (int j=n-2; j>=0; j--) {

            int max\_m=max(aux[i+1][j], aux[i][j+1]);

            aux[i][j]=max(mat[i][j], max\_m);

        }

    }

    int ans=-100001;

    for (int i=0; i<n-1; i++) {

        for (int j=0; j<n-1; j++) {

            ans=max(ans, aux[i+1][j+1]-mat[i][j]);

        }

    }

    return ans;

}