Farmer John's farm was flooded in the most recent storm, a fact only aggravated by the information that his cows are deathly afraid of water. His insurance agency will only repay him, however, an amount depending on the size of the largest "lake" on his farm.

The farm is represented as a rectangular grid with N (1 <= N <= 100) rows and M (1 <= M <= 100) columns. Each cell in the grid is either dry or submerged, and exactly K (1 <= K <= N\*M) of the cells are submerged. As one would expect, a lake has a central cell to which other cells connect by sharing a long edge (not a corner). Any cell that shares a long edge with the central cell or shares a long edge with any connected cell becomes a connected cell and is part of the lake.

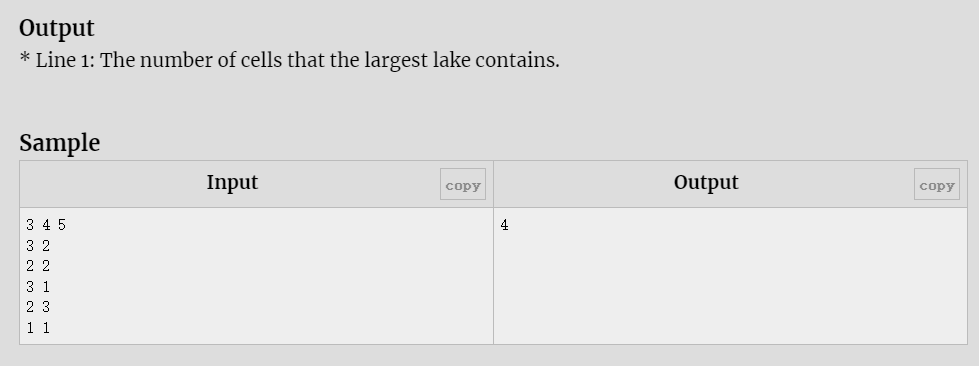
Input

\* Line 1: Three space-separated integers: N, M, and K

\* Lines 2..K+1: Line i+1 describes one submerged location with two space separated integers that are its row and column: R and C

Output

\* Line 1: The number of cells that the largest lake contains.



B - Substring and Subsequence

You are given two strings a*a* and b*b*, both consisting of lowercase Latin letters.

A *subsequence* of a string is a string which can be obtained by removing several (possibly zero) characters from the original string. A *substring* of a string is a contiguous subsequence of that string.

For example, consider the string abac:

* a, b, c, ab, aa, ac, ba, bc, aba, abc, aac, bac and abac are its subsequences;
* a, b, c, ab, ba, ac, aba, bac and abac are its substrings.

Your task is to calculate the minimum possible length of the string that contains a*a* as a substring and b*b* as a subsequence.

**Input**

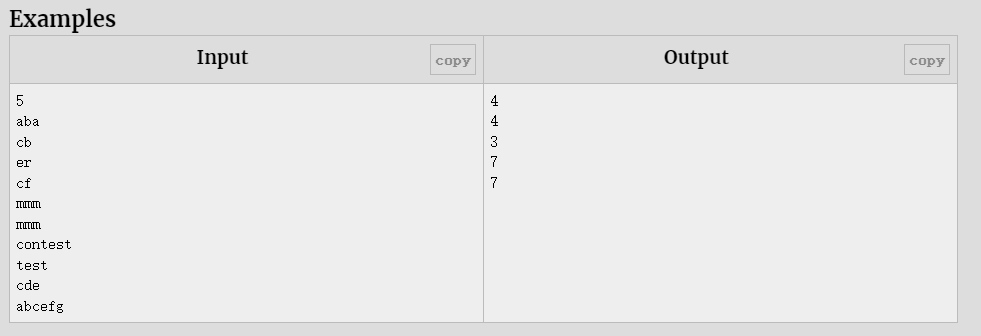
The first line contains a single integer t*t* (1≤t≤1031≤*t*≤103) — the number of test cases.

The first line of each test case contains a string a*a* (1≤∣a∣≤1001≤∣*a*∣≤100), consisting of lowercase Latin letters.

The second line of each test case contains a string b*b* (1≤∣b∣≤1001≤∣*b*∣≤100), consisting of lowercase Latin letters.

**Output**

For each test case, print a single integer — the minimum possible length of the string that contains a*a* as a substring and b*b* as a subsequence.



**Note**

In the examples below, the characters that correspond to the subsequence equal to b*b* are bolded.

In the first example, one of the possible answers is **c**a**b**a.

In the second example, one of the possible answers is er**cf**.

In the third example, one of the possible answers is **mmm**.

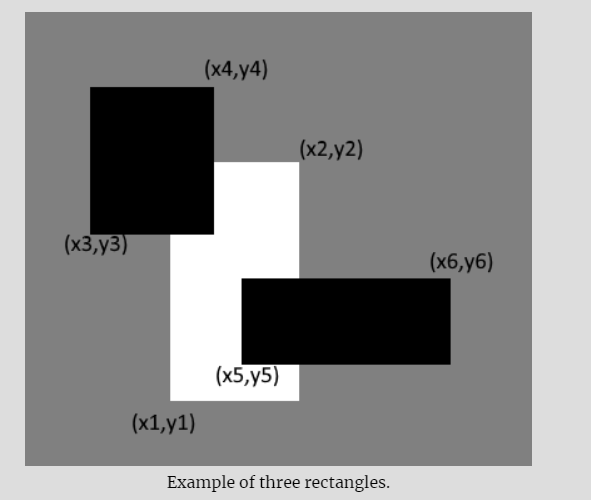
In the fourth example, one of the possible answers is con**test**.

In the fifth example, one of the possible answers is **abc**d**efg**.

C - White Sheet

There is a white sheet of paper lying on a rectangle table. The sheet is a rectangle with its sides parallel to the sides of the table. If you will take a look from above and assume that the bottom left corner of the table has coordinates (0,0)(0,0), and coordinate axes are left and bottom sides of the table, then the bottom left corner of the white sheet has coordinates (x1,y1)(*x*1​,*y*1​), and the top right — (x2,y2)(*x*2​,*y*2​).

After that two black sheets of paper are placed on the table. Sides of both black sheets are also parallel to the sides of the table. Coordinates of the bottom left corner of the first black sheet are (x3,y3)(*x*3​,*y*3​), and the top right — (x4,y4)(*x*4​,*y*4​). Coordinates of the bottom left corner of the second black sheet are (x5,y5)(*x*5​,*y*5​), and the top right — (x6,y6)(*x*6​,*y*6​).



Determine if some part of the white sheet can be seen from the above after the two black sheets are placed. The part of the white sheet can be seen if there is at least one point lying **not strictly inside** the white sheet and **strictly outside** of both black sheets.

**Input**

The first line of the input contains four integers x1,y1,x2,y2*x*1​,*y*1​,*x*2​,*y*2​ (0≤x1<x2≤106,0≤y1<y2≤106)(0≤*x*1​<*x*2​≤106,0≤*y*1​<*y*2​≤106) — coordinates of the bottom left and the top right corners of the white sheet.

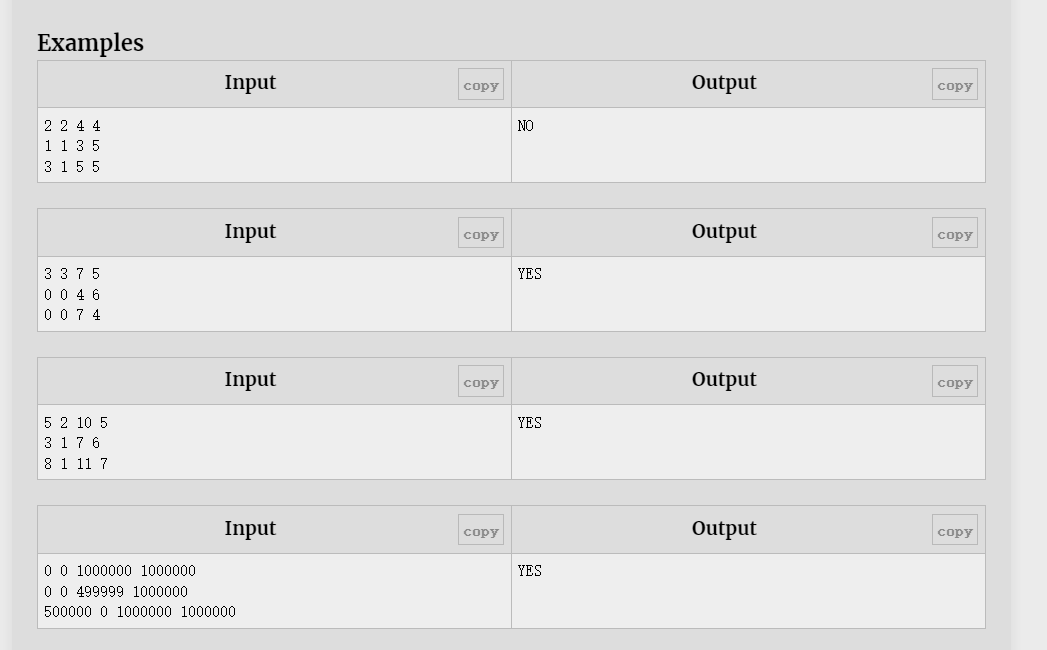
The second line of the input contains four integers x3,y3,x4,y4*x*3​,*y*3​,*x*4​,*y*4​ (0≤x3<x4≤106,0≤y3<y4≤106)(0≤*x*3​<*x*4​≤106,0≤*y*3​<*y*4​≤106) — coordinates of the bottom left and the top right corners of the first black sheet.

The third line of the input contains four integers x5,y5,x6,y6*x*5​,*y*5​,*x*6​,*y*6​ (0≤x5<x6≤106,0≤y5<y6≤106)(0≤*x*5​<*x*6​≤106,0≤*y*5​<*y*6​≤106) — coordinates of the bottom left and the top right corners of the second black sheet.

**The sides of each sheet of paper are parallel (perpendicular) to the coordinate axes.**

**Output**

If some part of the white sheet can be seen from the above after the two black sheets are placed, print "YES" (without quotes). Otherwise print "NO".



**Note**

In the first example the white sheet is fully covered by black sheets.

In the second example the part of the white sheet can be seen after two black sheets are placed. For example, the point (6.5,4.5)(6.5,4.5) lies not strictly inside the white sheet and lies strictly outside of both black sheets.

D - Bar Codes

A bar-code symbol consists of alternating dark and light bars, starting with a dark bar on the left. Each bar is a number of units wide. Figure 1 shows a bar-code symbol consisting of 4 bars that extend over 1+2+3+1=7 units.



In general, the bar code **BC(n, k, m)** is the set of all symbols with **k** bars that together extend over exactly **n** units, each bar being at most **m** units wide. For instance, the symbol in Figure 1 belongs to BC(7,4,3) but not to BC(7,4,2). Figure 2 shows all 16 symbols in BC(7,4,3). Each 1 represents a dark unit, each 0 represents a light unit.

0: 1000100 | 4: 1001110 | 8: 1100100 | 12: 1101110

1: 1000110 | 5: 1011000 | 9: 1100110 | 13: 1110010

2: 1001000 | 6: 1011100 | 10: 1101000 | 14: 1110100

3: 1001100 | 7: 1100010 | 11: 1101100 | 15: 1110110

Figure 2: All symbols of BC(7,4,3)

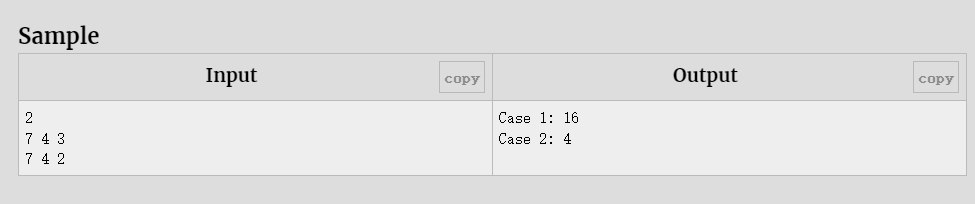
**Input**

Input starts with an integer **T (≤ 20000)**, denoting the number of test cases.

Each case contains three integers: **n, k, m (1 ≤ k, m ≤ n ≤ 50)**.

**Output**

For each case, print the case number and BC(n, k, m).



E - Space Shuttle Experiments

Professor Spook is consulting for NASA, which is planning a series of space shuttle flights and must decide which commercial experiments to perform and which instruments to have on board each flight. For each flight NASA considers a set **E = {E1, E2, …, Em}** of instruments experiments and the commercial sponsor of **Ej** has agreed to pay NASA **pj** dollars for the results of the experiments.

The experiments use a set **I = {I1, I2, …, In}** of instruments; each experiment **Ej** requires some of the instruments from the set. The cost of carrying instruments **Ik** is **ck** dollars. And an instrument can be used for multiple experiments.

The professor’s job is to determine which experiments to perform and which instruments to carry for a given flight in order to maximize the net revenue, which is the total income from the experiments performed minus the total cost of the instruments carried. Since he is not a programmer, he asked your help.

**Input**

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case starts with a line containing two integers **m (1 ≤ m ≤ 100)** and \*\*n (1 ≤ n ≤ 100), \*\*where **m** denotes the number of experiments and **n** denotes the number of instruments. The next line contains **m** space separated integers, where the **jth** integer denotes the commercial sponsor of **Ej** paying NASA **pj(1 ≤ pj ≤ 10000)** dollars for the result of the experiment. The next line contains **n** space separated integers, where the **kth** integer denotes the cost of carrying the **kth** instrument, **ck(1 ≤ ck ≤ 10000)**. Each of the next **m** lines contains an integer **qi (1 ≤ qi ≤ n)** followed by **qi** distinct integers each between **1** and **n**, separated by spaces. These **qi** integers denote the required instruments for the **ith** experiment.

**Output**

For each case, print the case number and the maximum revenue NASA can make using the experiments.

