第 36 届 ACM 国际大学生程序设计竞赛

亚洲区上海预选赛

The 36th ACM/ICPC Asia Regional Shanghai Site

Online Contest

A. 24 Puzzle

Daniel likes to play a special board game, called 24 puzzle. 24 puzzle is such a game that there are tiles with the number 1 to 23 in a play board like the follow picture:

```
 #  #
######
 ####
 ####
######
 #  #
```

The '#' denotes the positions that the tiles may be placed on. There are 24 possible positions in total, so one of them is not occupied by the tile. We can denote the empty position by zero.

Daniel could move the tiles to the empty position if the tile is on the top, bottom, left or right of the empty position. In this way Daniel can reorder the tiles on the board.

Usually he plays with this game by setting up a target states initially, and then trying to do a series of moves to achieve the target. Soon he finds that not all target states could be achieved.

He asks for your help, to determine whether he has set up an impossible target or not.

Input

The first line of input contains an integer denoting the number of test cases.

For each test case, the first line contains 24 integers denoting the initial states of the game board. The numbers are the describing the tiles from top to bottom, left to right. And the empty position is indicated by zero. You can assume that the number of each tile are different, and there must be exactly one empty position. The second line of test case also contains 24 integers denoting the target states.

Output

For each test case, if the target is impossible to achieve, output 'Y' in a single line, otherwise, output 'N'.

Sample Input

```
2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
3 1 2 0 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
3 0 2 1 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

Sample Output

```
N
Y
```

B. Bombing

It's a cruel war which killed millions of people and ruined series of cities. In order to stop it, let's bomb the opponent's base.

It seems not to be a hard work in circumstances of street battles, however, you'll be encountered a much more difficult instance: recounting exploits of the military. In the bombing action, the commander will dispatch a group of bombers with weapons having the huge destructive power to destroy all the targets in a line. Thanks to the outstanding work of our spy, the positions of all opponents' bases had been detected and marked on the map, consequently, the bombing plan will be sent to you.

Specifically, the map is expressed as a 2D-plane with some positions of enemy's bases marked on. The bombers are dispatched orderly and each of them will bomb a vertical or horizontal line on the map. Then your commanded wants you to report that how many bases will be destroyed by each bomber. Notice that a ruined base will not be taken into account when calculating the exploits of later bombers.

Input

Multiple test cases and each test cases starts with two non-negative integer N (N<=100,000) and M (M<=100,000) denoting the number of target bases and the number of scheduled bombers respectively. In the following N line, there is a pair of integers x and y separated by single space indicating the coordinate of position of each opponent's base. The following M lines describe the bombers, each of them contains two integers c and d where c is 0 or 1 and d is an integer with absolute value no more than 10^9, if c = 0, then this bomber will bomb the line x = d, otherwise y = d. The input will end when N = M = 0 and the number of test cases is no more than 50.

Output

For each test case, output M lines, the ith line contains a single integer denoting the number of bases that were destroyed by the corresponding bomber in the input. Output a blank line after each test case.

Sample Input

```
3 2
1 2
1 3
2 3
0 1
1 3
0 0
```

Sample Output

2
1

C. Game

Alice and Bob are playing game with each other. They play the game on a 2D board. Alice has many vertical 1*2 tiles while Bob has many horizontal 2*1 tiles. They take turn to place their own tiles on the board. Considering about that the tiles cannot overlap each other, the player cannot do the placement any more loses. Since this is such a complex game that they could not find optimal method to play that, Alice decide to simplify this game by replace the large 2D board by some small ones. Alice set up a lot of Tetris tiles instead of the original 2D board. In the other words, the player can only place their own vertical or horizontal tiles on the Tetris-like board. Each player can choose one possible place on any Tetris tiles to place its own tiles. In fact, there are following 15 types of Tetris playground.

```
(1)  (2)  (3)   (4)  (5)(6)  (7)  (8)(9)  (10)  (11)(12)(13)  (14)  (15)
#    #### ##     ##  #     #  #    #  ###   ###  ###    #   #    #    ##
#          ##   ##   ##   ##  #    #  #       #    #   ##  ##   ###   ##
#                     #   #   ##  ##                    #   #
#
```

The playground cannot be transformed in any ways, including reflection and rotation.

Given the number of each type of tiles, you are asked to determine who will win the game if Alice plays first and both players are playing optimal.

Input

There are multiple test cases; the first line of input contains a single integer denoting the number of test cases.

For each test case, there are only one line contains 15 integers denoting the number of Tetris tiles of the above 15 types. All the numbers are no greater than 100.

Output

For each test cases, output “Alice” if Alice will win the game and both player plays optimally, “Bob” otherwise.

Sample Input

```
3
5 4 0 0 0 0 0 0 0 0 0 0 0 0 0
5 5 0 0 0 0 0 0 0 0 0 0 0 0 0
100 100 0 0 0 0 0 0 0 0 0 2 1 0 0
```

Sample Output

```
Case #1: Alice
```

```
Case #2: Bob
Case #3: Alice
```

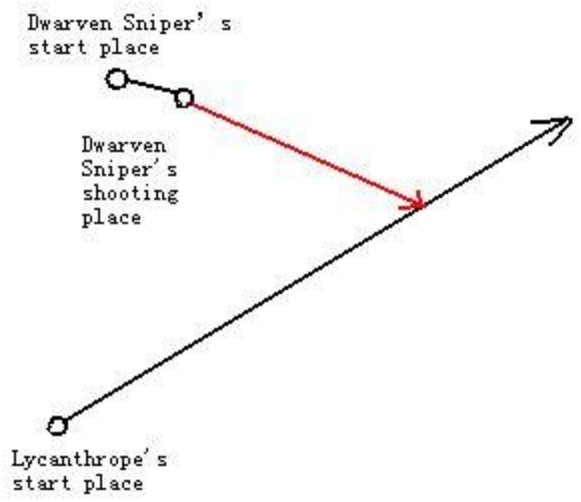
D. Dwarven Sniper's hunting

Now the hunting starts in the world named DOTA, a stupid PC game which cannot play with others together.

Among the individuals in the game, there are two heroes named Dwarven Sniper and Lycanthrope. Lycanthrope wants to escape from being captured; however, our Dwarven Sniper won't let him go! He will use the Silver Bullet to kill the Lycanthrope by only one shot! Yes, that's enough.

Lycanthrope is running on a line in the map with a constant speed and direction. The weapon range of the Silver Bullet is limited by L meters. Dwarven Sniper can run for a while freely, and then shoot Lycanthrope. In order to show his excellent shooting skill, Dwarven Sniper wants the Silver Bullet flying as far as possible. But don't forget the flying time of the Silver Bullet due to considerable weight of the bullet. And Dwarven Sniper wants to stop the hunting as quickly as possible. So if there is more than one way to show his excellent skill, he would choose the fastest way. In this problem we consider the Silver Bullet and Lycanthrope as two points.

Now Dwarven Sniper wants to know the maximum length that the Silver Bullet can fly, and the shortest time that the hunting lasts. Specifically, the total hunting time is defined as the time interval from the start of hunting to the moment that the bullet hit Lycanthrope. Can you help him?



Input

There are several test cases. Each of them contains only one line which consist of 9 real numbers, that are X1, Y1, X2, Y2, Lx, Ly, vD , vB and L (-10000 <= X1 , Y1 , X2 , Y2 , Lx , Ly <= 10000 , 0 <= vD , vB , L <=100000).

The pair (X1, Y1) is the starting position of the Lycanthrope while (X2, Y2) is the starting position of Dwarven Sniper.

(Lx, Ly) is the moving vector per second of the Lycanthrope.

vD is the speed of the Dwarven Sniper .

vB is the speed of the Silver Bullet.

All units are meters/second.

It is guaranteed that (Lx*Lx+Ly*Ly) < vD*vD < vB*vB , and Dwarven Sniper's starting position is different from Lycanthrope's position. The input ends with a line containing all zeros.

Output

For each test case, output two real numbers S and T in a line separated by a single space denoting that the Silver bullet flies S meters before hitting Lycanthrope and the hunting lasts for T seconds, both with 3 digits after the decimal point.

You may assume that Dwarven Sniper can finish his hunting within no more than 1e+9 seconds.

Sample Input

```
-1 0 0 10 1 0 2 10 10
0 0 0 5 0 1 2 6 6
0 0 0 0 0 0 0 0 0
```

Sample Output

```
10.000 1.000
6.000 3.000
```

E. Equation of XOR

Recently, Jimmy is learning about linear algebra from Blue Mary while having the course of Boolean algebra in class offered by Prof. Z. Since Jimmy has been thoroughly bored by the boring homework assigned by two teachers, evil Jimmy plans to set a hard question to baffle them as revenge for their heavy tasks. As a result, Jimmy comes up with an idea that merging the knowledge from both the two classes and constructs a complicate problem: the XOR equation system.

Let's consider the following equations:

$$(a11 \bullet x1) \wedge (a12 \bullet x2) \wedge \ldots \wedge (a1m \bullet xm) = 0$$
$$(a21 \bullet x1) \wedge (a22 \bullet x2) \wedge \ldots \wedge (a2m \bullet xm) = 0$$
$$\ldots$$
$$(an1 \bullet x1) \wedge (an2 \bullet x2) \wedge \ldots \wedge (anm \bullet xm) = 0$$

which satisfies the following conditions:

1. aij in {0,1} for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$;
2. xi in Si where Si is a subset of {0,1,2,3}, $1 \leqslant i \leqslant m$;
3. $|Si| \leqslant 3$, $1 \leqslant i \leqslant m$;
4. $1 \leqslant n \leqslant 30$, $1 \leqslant m \leqslant 22$.

In the system of equations, operation " • " denotes the multiplication operation while " ^ " is for bitwise XOR. Moreover, the bitwise XOR takes two bit patterns of equal length and performs the logical XOR operation on each pair of corresponding bits. The result in each position is 1 if the two bits are different, and 0 if they are the same.

Rather than expecting a solution of a specified equation system, Jimmy would like to ask the teachers to calculate that how many distinct solutions can satisfy a given equation system. What a confusing puzzle! Help Jimmy's teachers please!

Input

There are several test cases. The first line of input is a single positive integer T (<= 15) indicating the number of test cases, then T cases follow.

For each test case, the first line contains two integers N and M giving the two dimensions of the equation system respectively where N is the number of rows and M for columns. Then N lines are following, each line contains m integers. Item at line i and column j represents aij. The next m lines are descriptions of Si that the leading integer K denotes the number of elements in Si and the following K integers represent elements.

Output

For each test case, output one line containing a single integer which represents the answer of this case.

Sample Input

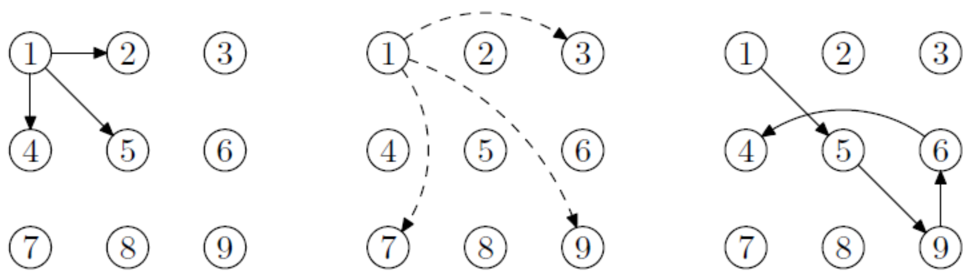
1
1 2
1 1
3 0 1 2
2 3 1

Sample Output

1

F. Unlock the Cell Phone

Modern high-tech cell phones use unlock patterns to unlock the system. The pattern is usually a 3*3 dot array. By moving your finger over there dots, you can generate your personal unlock pattern. More specifically, press your finger over any starting dot, then slide all the way to the next dot, touch it, and so on. Jumping is not allowed. For example, starting from dot 1, you can slide to touch dot 2, dot 4 and dot 5, but sliding directly to dot 3, dot 7 or dot 9 are not allowed. Note that sliding from 1 to 6 and 8 is also allowed because they are not considered as jumping over any dot. However, you can jump a dot if it has been touched before. For example, staring with 1-5-9-6, you can slide directly to dot 4.



Here is a very particular cell phone. It has a dot array of size n*m. Some of the dots are ordinary ones: you can touch, and slide over them when touched before; some are forbidden ones: you cannot touch or slide over them; some are inactive ones: you cannot touch them, but can slide over them. Each dot can only be touched once. You are required to calculate how many different unlock patterns passing through all the ordinary dots.

Input

The input contains several test cases. Each test case begins with a line containing two integers n and m (1 <= n, m <= 5), indicating the row and column number of the lock keypad. The following n lines each contains m integers kij indicating the properties of each key, kij=0 stands for an ordinary key, kih=1 stands for a forbidden key; and kij=2 stands for an inactive key. The number of ordinary keys is greater than zero and no more than 16.

Output

For each test, output an integer indicating the number of different lock patterns.

Sample Input

```
2 2
0 0
0 0
3 3
0 0 0
```

0 2 1
0 0 0

Sample Output

24
2140

G. Can you answer these queries?

A lot of battleships of evil are arranged in a line before the battle. Our commander decides to use our secret weapon to eliminate the battleships. Each of the battleships can be marked a value of endurance. For every attack of our secret weapon, it could decrease the endurance of a consecutive part of battleships by make their endurance to the square root of its original value of endurance. During the series of attack of our secret weapon, the commander wants to evaluate the effect of the weapon, so he asks you for help.

You are asked to answer the queries that the sum of the endurance of a consecutive part of the battleship line.

Input

The input contains several test cases, terminated by EOF.

For each test case, the first line contains a single integer N, denoting there are N battleships of evil in a line. (1 <= N <= 100000)

The second line contains N integers Ei, indicating the endurance value of each battleship from the beginning of the line to the end. You can assume that the sum of all endurance value is less than 2^{63}.

The next line contains an integer M, denoting the number of actions and queries. (1 <= M <= 100000)

For the following M lines, each line contains three integers T, X and Y. The T=0 denoting the action of the secret weapon, which will decrease the endurance value of the battleships between the X-th and Y-th battleship, inclusive. The T=1 denoting the query of the commander which ask for the sum of the endurance value of the battleship between X-th and Y-th, inclusive.

Output

For each test case, print the case number at the first line. Then print one line for each query. <u>And remember follow a blank line after each test case.</u>

Sample Input

```
10
1 2 3 4 5 6 7 8 9 10
5
0 1 10
1 1 10
1 1 5
0 5 8
1 4 8
```

Sample Output

```
Case #1:
19
7
6
```

H. The time of a day

There are no days and nights on byte island, so the residents here can hardly determine the length of a single day. Fortunately, they have invented a clock with several pointers. They have N pointers which can move round the clock. Every pointer ticks once per second and the i-th pointer move to the starting position after i times of ticks. The wise of the byte island decide to define a day as the time interval between the initial time and the first time when all the pointers moves to the position exactly the same as the initial time.

The wise of the island decide to choose some of the N pointers to make the length of the day greater or equal to M. They want to know how many different ways there are to make it possible.

Input

There are a lot of test cases. The first line of input contains exactly one integer, indicating the number of test cases.

For each test cases, there are only one line contains two integers N and M, indicating the number of pointers and the lower bound for seconds of a day M. ($1 <= N <= 40$, $1 <= M <= 2^{63}-1$)

Output

For each test case, output a single integer denoting the number of ways.

Sample Input

```
3
5 5
10 1
10 128
```

Sample Output

```
Case #1: 22
Case #2: 1023
Case #3: 586
```

I. Distinct Sub-matrix

In this problem, let us consider an N*M matrix of capital letters. By selecting consecutive columns and rows, we can define the sub-matrix as the elements on chosen columns and rows.

Two sub-matrices should be regarded the same if and only if they have the same detentions and characters (which, of course, are capital letters) on corresponding position. It is your task to find the number of distinct sub-matrices of a given letter matrix.

Input

The input contains a lot of test cases. The first line of input contains exactly one integer, indicating the number of test cases.

For each of the test case, the first line contains two integers N and M, denoting the number of rows and columns of the given matrix. (1 <= N, M <= 128)

The next N lines contain only capital letters, indicating the given matrix.

Output

For each test case, output a single integer denoting the number of distinct sub-matrices.

Sample Input

```
2
2 2
AB
BA
3 3
ABA
BAA
AAA
```

Sample Output

```
Case #1: 7
Case #2: 22
```

J. Tank

Surprisingly, the king of the Quadruple Island Kingdom suddenly declares war on its neighbor country, the Utopia Land. After that, the King dispatches a large number of tanks for first attack. You, as the Defense Minister of Utopia Land, are certainly under mass of pressure. In order to defense the enemy, you decide to use the new secret weapon: laser cannon. This kind of weapon is so powerful that can easily destroy any thing at a line in a moment, or course, including tanks.
However, Because of some designing flaw, you can use such weapon just once during the war. The only hope is that you can choose a good moment and right way to use it so that as many tanks as possible can be destroyed.

A tank can be described as a point whose coordinate is (Xi, Yi) on a 2D-plain with speed vector (VXi, VYi) per second. Starting at time 0, you can use your laser cannon at any position and in any direction at any INTEGER number of seconds after time 0. It is possible that some tanks will meet at the same point while moving, and they will not influence each other anyway.

Input

There are multiple test cases and the number of test cases is no more than 100. For each test cases, The first line contains a single integer N (1 <= N <= 100) indicating the number of tanks. Then N lines are following that describing the tanks respectively. Each of them consists of four nonzero integer Xi, Yi, VXi and VYi separated by single space where |Xi|,|Yi| <= 100,000,000 and |VXi|,|VYi| <= 100 denoting the initial position of a tank and the speed vector. Input will end when N = 0.

Output

For each test case, output one line with a single integer indicating the largest number of tanks which can be destroyed by the laser cannon.

Sample Input

```
4
10 10 -1 -1
3 1 4 6
2 8 5 -1
1 3 1 4
0
```

Sample Output

```
3
```