

There is a curious man called Matt.

One day, Matt's best friend Ted is wandering on the non-negative half of the number line. Matt finds it interesting to know the maximal speed Ted may reach. In order to do so, Matt takes records of Ted's position. Now Matt has a great deal of records. Please help him to find out the maximal speed Ted may reach, assuming Ted moves with a constant speed between two consecutive records.

Input

The first line contains only one integer T , which indicates the number of test cases.

For each test case, the first line contains an integer N ($2 \leq N \leq 10000$), indicating the number of records.

Each of the following N lines contains two integers t_i and x_i ($0 \leq t_i, x_i \leq 10^6$), indicating the time when this record is taken and Ted's corresponding position. Note that records may be unsorted by time. It's guaranteed that all t_i would be distinct.

Output

For each test case, output a single line 'Case # x : y ', where x is the case number (starting from 1), and y is the maximal speed Ted may reach. The result should be rounded to two decimal places.

Hint:

In the first sample, Ted moves from 2 to 4 in 1 time unit. The speed $2/1$ is maximal.

In the second sample, Ted moves from 5 to 0 in 1 time unit. The speed $5/1$ is maximal.

Sample Input

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

Sample Output

```
Case #1: 2.00
Case #2: 5.00
```

In mathematics, the four color theorem, or the four color map theorem, states that, given any separation of a plane into contiguous regions, producing a figure called a map, no more than four colors are required to color the regions of the map so that no two adjacent regions have the same color.

— Wikipedia, the free encyclopedia

In this problem, you have to solve the 4-color problem. Hey, I'm just joking.

You are asked to solve a similar problem:

Color an $N \times M$ chessboard with K colors numbered from 1 to K such that no two adjacent cells have the same color (two cells are adjacent if they share an edge). The i -th color should be used in exactly c_i cells.

Matt hopes you can tell him a possible coloring.

Input

The first line contains only one integer T ($1 \leq T \leq 5000$), which indicates the number of test cases.

For each test case, the first line contains three integers: N, M, K ($0 < N, M \leq 5, 0 < K \leq N * M$).

The second line contains K integers c_i ($c_i > 0$), denoting the number of cells where the i -th color should be used.

It's guaranteed that $c_1 + c_2 + \dots + c_K = N * M$.

Output

For each test case, the first line contains 'Case # x :', where x is the case number (starting from 1).

In the second line, output 'NO' if there is no coloring satisfying the requirements. Otherwise, output 'YES' in one line. Each of the following N lines contains M numbers seperated by single whitespace, denoting the color of the cells.

If there are multiple solutions, output any of them.

Sample Input

4
1 5 2
4 1
3 3 4
1 2 2 4
2 3 3
2 2 2
3 2 3
2 2 2

Sample Output

Case #1:
NO
Case #2:
YES
4 3 4
2 1 2
4 3 4
Case #3:
YES
1 2 3
2 3 1
Case #4:
YES
1 2
2 3
3 1

Matt is playing a naive computer game with his deeply loved pure girl.

The playground is a rectangle with walls around. Two balls are put in different positions inside the rectangle. The balls are so tiny that their volume can be ignored. Initially, two balls will move with velocity $(1, 1)$. When a ball collides with any side of the rectangle, it will rebound without loss of energy. The rebound follows the law of reflection (i.e. the angle at which the ball is incident on the wall equals the angle at which it is reflected).

After they choose the initial position, Matt wants you to tell him where will the two balls collide for the first time.

Input

The first line contains only one integer T which indicates the number of test cases.

For each test case, the first line contains two integers x and y . The four vertices of the rectangle are $(0, 0)$, $(x, 0)$, $(0, y)$ and (x, y) . ($1 \leq x, y \leq 10^5$)

The next line contains four integers x_1, y_1, x_2, y_2 . The initial position of the two balls is (x_1, y_1) and (x_2, y_2) . ($0 \leq x_1, x_2 \leq x; 0 \leq y_1, y_2 \leq y$)

Output

For each test case, output 'Case #x:' in the first line, where x is the case number (starting from 1).

In the second line, output 'Collision will not happen.' (without quotes) if the collision will never happen. Otherwise, output two real numbers x_c and y_c , rounded to one decimal place, which indicate the position where the two balls will first collide.

Hint: In first example, two balls move from $(1, 1)$ and $(9, 9)$ both with velocity $(1, 1)$, the ball starts from $(9, 9)$ will rebound at point $(10, 10)$ then move with velocity $(-1, -1)$. The two balls will meet each other at $(6, 6)$.

Sample Input

```
3
10 10
1 1 9 9
10 10
0 5 5 10
10 10
1 0 1 10
```

Sample Output

```
Case #1:
6.0 6.0
Case #2:
Collision will not happen.
Case #3:
6.0 5.0
```

Dire wolves, also known as Dark wolves, are extraordinarily large and powerful wolves. Many, if not all, Dire Wolves appear to originate from Draenor.

Dire wolves look like normal wolves, but these creatures are of nearly twice the size. These powerful beasts, 8 - 9 feet long and weighing 600 - 800 pounds, are the most well-known orc mounts. As tall as a man, these great wolves have long tusked jaws that look like they could snap an iron bar. They have burning red eyes. Dire wolves are mottled gray or black in color. Dire wolves thrive in the northern regions of Kalimdor and in Mulgore.

Dire wolves are efficient pack hunters that kill anything they catch. They prefer to attack in packs, surrounding and flanking a foe when they can.

— Wowpedia, Your wiki guide to the World of Warcraft

Matt, an adventurer from the Eastern Kingdoms, meets a pack of dire wolves. There are N wolves standing in a row (numbered with 1 to N from left to right). Matt has to defeat all of them to survive.

Once Matt defeats a dire wolf, he will take some damage which is equal to the wolf's current attack. As gregarious beasts, each dire wolf i can increase its adjacent wolves' attack by b_i . Thus, each dire wolf i 's current attack consists of two parts, its basic attack a_i and the extra attack provided by the current adjacent wolves. The increase of attack is temporary. Once a wolf is defeated, its adjacent wolves will no longer get extra attack from it. However, these two wolves (if exist) will become adjacent to each other now.

For example, suppose there are 3 dire wolves standing in a row, whose basic attacks a_i are (3, 5, 7), respectively. The extra attacks b_i they can provide are (8, 2, 0). Thus, the current attacks of them are (5, 13, 9). If Matt defeats the second wolf first, he will get 13 points of damage and the alive wolves' current attacks become (3, 15).

As an alert and resourceful adventurer, Matt can decide the order of the dire wolves he defeats. Therefore, he wants to know the least damage he has to take to defeat all the wolves.

Input

The first line contains only one integer T , which indicates the number of test cases. For each test case, the first line contains only one integer N ($2 \leq N \leq 200$).

The second line contains N integers a_i ($0 \leq a_i \leq 100000$), denoting the basic attack of each dire wolf.

The third line contains N integers b_i ($0 \leq b_i \leq 50000$), denoting the extra attack each dire wolf can provide.

Output

For each test case, output a single line 'Case # x : y ', where x is the case number (starting from 1), y is the least damage Matt needs to take.

Hint: In the first sample, Matt defeats the dire wolves from left to right. He takes $5 + 5 + 7 = 17$ points of damage which is the least damage he has to take.

Sample Input

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

Sample Output

```
Case #1: 17
Case #2: 74
```

Matt loves letter L.

A point set P is (a,b) -L if and only if there exists x, y satisfying:

$$P = \{(x,y), (x+1,y), \dots, (x+a,y), (x,y+1), \dots, (x,y+b)\} \quad (a,b \geq 1)$$

A point set Q is good if and only if Q is an (a,b) -L set and $\gcd(a,b) = 1$.

Matt is given a point set S . Please help him find the number of ordered pairs of sets (A,B) such that:

- A, B are both *good*,
- $A, B \subseteq S$ and $A \cap B = \emptyset$

Input

The first line contains only one integer T , which indicates the number of test cases.

For each test case, the first line contains an integer N ($0 \leq N \leq 40000$), indicating the size of the point set S .

Each of the following N lines contains two integers x_i, y_i , indicating the i -th point in S ($1 \leq x_i, y_i \leq 200$). It's guaranteed that all (x_i, y_i) would be distinct.

Output

For each test case, output a single line ‘Case # x : y ’, where x is the case number (starting from 1) and y is the number of pairs.

Hint: In the second sample, the ordered pairs of sets Matt can choose are:

- $A = \{(1,1), (1,2), (1,3), (2,1)\}$ and $B = \{(2,2), (2,3), (3,2)\}$
- $A = \{(2,2), (2,3), (3,2)\}$ and $B = \{(1,1), (1,2), (1,3), (2,1)\}$
- $A = \{(1,1), (1,2), (2,1), (3,1)\}$ and $B = \{(2,2), (2,3), (3,2)\}$
- $A = \{(2,2), (2,3), (3,2)\}$ and $B = \{(1,1), (1,2), (2,1), (3,1)\}$
- $A = \{(1,1), (1,2), (2,1)\}$ and $B = \{(2,2), (2,3), (3,2)\}$
- $A = \{(2,2), (2,3), (3,2)\}$ and $B = \{(1,1), (1,2), (2,1)\}$

Hence, the answer is 6.

Sample Input

2
6
1 1
1 2
2 1
3 3
3 4
4 3
9
1 1
1 2
1 3
2 1
2 2
2 3
3 1
3 2
3 3

Sample Output

Case #1: 2
Case #2: 6

Matt, a famous adventurer who once defeated a pack of dire wolves alone, found a lost court. Matt finds that there are N fluorescent lights which seem to be the stars from the firmament. What's more, there are M switches that control these fluorescent lights. Each switch is connected to a group of lights. When Matt touches a switch, all the lights connected to it will change their states (turning the dark on, turning the bright off).

Initially, all the fluorescent lights are dark. For each switch, Matt will touch it with probability 1. As a curious gentleman, Matt wants to calculate $E[X^3]$, where X represents the number of bright lights at the end, $E[X^3]$ represents the expectation of cube of X .

Input

The first line contains only one integer T , which indicates the number of test cases.

For each test case, the first line contains N, M ($1 \leq N, M \leq 50$), denoting the number of fluorescent lights (numbered from 1 to N) and the number of switches (numbered from 1 to M).

M lines follow. The i -th line begins with an integer K_i ($1 \leq K_i \leq N$). K_i distinct integers l_{ij} ($1 \leq l_{ij} \leq N$) follow, denoting the fluorescent lights that the i -th switch controls.

Output

For each test case, output a single line 'Case # x : y ', where x is the case number (starting from 1) and y is the answer. To avoid rounding error, the answer you should output is:

$$E[X^3] \times 2^M \bmod (10^9 + 7)$$

Hint:

- For the **first** sample, there're 4 possible situations:
- All the switches is off, no light is bright, $X^3 = 0$.
 - Only the first switch is on, the first light is bright, $X^3 = 1$.
 - Only the second switch is on, all the lights are bright, $X^3 = 8$.
 - All the switches is on, the second lights are bright, $X^3 = 1$.

Therefore, the answer is $E[X^3] \times 2^2 \bmod (10^9 + 7) = 10$.

- For the **second** sample, there're 2 possible situations:
- The switches is off, no light is bright, $X^3 = 0$.
 - The switches is on, all the lights are bright, $X^3 = 27$.

Therefore, the answer is $E[X^3] \times 2^1 \bmod (10^9 + 7) = 27$.

Sample Input

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

Sample Output

```
Case #1: 10
Case #2: 27
```

Now Matt is preparing for the Graduate Record Examinations as Coach Pang did in 2013 and George did in 2011.

Thanks to modern techniques, Matt uses automata instead of old-fashioned vocabulary books. The automata used by Matt is a directed acyclic graph (DAG) with N vertices and M edges. The vertices are conveniently numbered by $1, 2, \dots, N$. Each edge is labeled with an integer. Additionally, some vertices are marked as special. A GRE word is obtained by concatenating the labels on the path from vertex 1 to a special vertex. Now, Matt has Q questions. The i -th question is asking for the length of k_i -th smallest words among all the GRE words he can obtain in lexicographical order.

Input

The first line contains only one integer T , which indicates the number of test cases. For each test case, the first line contains three integers N, M, Q ($2 \leq N \leq 10^5, 0 \leq M \leq 10^5, 1 \leq Q \leq 10^5$). The second line contains $N - 1$ integers s_2, \dots, s_n . If the i -th vertex is special, then $s_i = 1$. Otherwise, $s_i = 0$. Vertex 1 is never special. Each of the following M lines contains three integers a_i, b_i, c_i denoting an edge from vertex a_i to vertex b_i labeled with c_i ($1 \leq a_i, b_i \leq N, 1 \leq c_i \leq 10^9$). For each vertex v , all outgoing edges are labeled with distinct integers. Each of the following Q lines contains the integer k_i ($1 \leq k_i \leq 10^8$) of the i -th question.

Output

For each test case, output ‘Case # x :’ in the first line, where x is the case number (starting from 1). Then, for each question, output the length of the word in one line. If the word does not exist, output ‘-1’ (without quotes) instead. **Hint:** There are 3 GRE words in total (sorted in lexicographical order):

- 1. (1)
- 2. (1, 3)
- 3. (12)

Sample Input

```
1
3 3 4
1 1
1 2 1
1 3 12
2 3 3
1
2
3
4
```

Sample Output

```
Case #1:
1
2
1
-1
```

Matt has N friends. They are playing a game together.

Each of Matt's friends has a magic number. In the game, Matt selects some (could be zero) of his friends. If the **xor** (exclusive-or) sum of the selected friend's magic numbers is no less than M , Matt wins.

Matt wants to know the number of ways to win.

Input

The first line contains only one integer T , which indicates the number of test cases.

For each test case, the first line contains two integers N, M ($1 \leq N \leq 40, 0 \leq M \leq 10^6$).

In the second line, there are N integers k_i ($0 \leq k_i \leq 10^6$), indicating the i -th friend's magic number.

Output

For each test case, output a single line 'Case # x : y ', where x is the case number (starting from 1) and y indicates the number of ways where Matt can win.

Hint: In the first sample, Matt can win by selecting:

- friend with number 1 and friend with number 2. The **xor** sum is 3.
- friend with number 1 and friend with number 3. The **xor** sum is 2.
- friend with number 2. The **xor** sum is 2.
- friend with number 3. The **xor** sum is 3. Hence, the answer is 4.

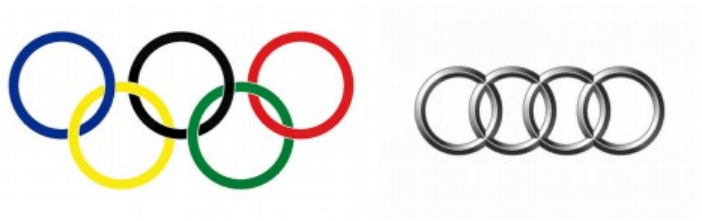
Sample Input

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

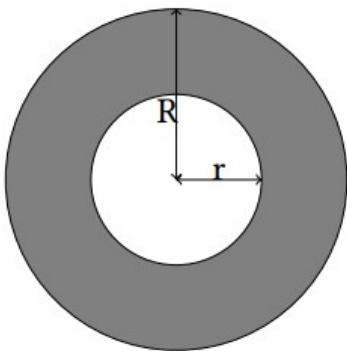
Sample Output

```
Case #1: 4
Case #2: 2
```


Matt is a big fan of logo design. Recently he falls in love with logo made up by rings. The following figures are some famous examples you may know.



A ring is a 2-D figure bounded by two circles sharing the common center. The radius for these circles are denoted by r and R ($r < R$). For more details, refer to the gray part in the illustration below.



Matt just designed a new logo consisting of two rings with the same size in the 2-D plane. For his interests, Matt would like to know the area of the intersection of these two rings.

Input

The first line contains only one integer T ($T \leq 10^5$), which indicates the number of test cases. For each test case, the first line contains two integers r, R ($0 \leq r < R \leq 10$).

Each of the following two lines contains two integers x_i, y_i ($0 \leq x_i, y_i \leq 20$) indicating the coordinates of the center of each ring.

Output

For each test case, output a single line ‘Case # x : y ’, where x is the case number (starting from 1) and y is the area of intersection rounded to 6 decimal places.

Sample Input

2
2 3
0 0
0 0
2 3
0 0
5 0

Sample Output

Case #1: 15.707963
Case #2: 2.250778

As we all know, Matt is an outstanding contestant in ACM-ICPC. Graph problems are his favorite. Once, he came up with a simple algorithm for finding the maximal independent set in trees by mistake.

A tree is a connected undirected graph without cycles, and an independent set is subset of the vertex set which contains no adjacent vertex pairs.

Suppose that the tree contains N vertices, conveniently numbered by $1, 2, \dots, N$. First, Matt picks a permutation p_1, p_2, \dots, p_N of $\{1, 2, 3, \dots, N\}$ randomly and uniformly.

After picking the permutation, Matt does the following procedure.

1. Set $S = \emptyset$.
2. Consider the vertex p_1, p_2, \dots, p_N accordingly. For vertex p_i , if and only if there is no vertex in S which is adjacent to p_i , add vertex p_i into S .
3. Output the set S .

Clearly the above algorithm does not always output the maximal independent set. Matt would like to know the expected size of set S instead.

Input

The first line contains only one integer T , which indicates the number of test cases.

For each test case, the first line contains an integer N ($1 \leq N \leq 200$), indicating the number of vertices in the graph.

Each of the following $N - 1$ lines contains two integers u, v ($1 \leq u, v \leq N$) indicating an edge between u and v . You may assume that all the vertices are connected.

Output

For each test case, output a single line ‘Case # x : y ’, where x is the case number (starting from 1) and y is the answer. To avoid rounding error, the answer you should output is:

$$(\text{the expected size of independent set}) \times N! \bmod (10^9 + 7)$$

Hint: In the first sample, there are 4 vertices, so there are $4!$ permutations Matt may get.

Suppose the permutation Matt gets is 1 2 3 4. He will add vertex 1 into the independent set.

Suppose the permutation Matt gets is 2 1 3 4. He will add vertex 2, vertex 3 and vertex 4 into the independent set.

It is obvious that if the first element in the permutation is not vertex 1, he will get an independent set whose size is 3. Otherwise, he will get an independent set whose size is 1. Since there are 18 permutations whose first element is not vertex 1, the answer in the first sample is $(3 \times 18 + 1 \times 6) \bmod (10^9 + 7) = 60$.

Sample Input

```
2
4
1 2
1 3
1 4
3
1 2
2 3
```

Sample Output

```
Case #1: 60
Case #2: 10
```

Matt's friend K.Bro is an ACMer.

Yesterday, K.Bro learnt an algorithm: Bubble sort. Bubble sort will compare each pair of adjacent items and swap them if they are in the wrong order. The process repeats until no swap is needed.

Today, K.Bro comes up with a new algorithm and names it K.Bro Sorting.

There are many rounds in K.Bro Sorting. For each round, K.Bro chooses a number, and keeps swapping it with its next number while the next number is less than it.

For example, if the sequence is "1 4 3 2 5", and K.Bro chooses "4", he will get "1 3 2 4 5" after this round. K.Bro Sorting is similar to Bubble sort, but it's a randomized algorithm because K.Bro will choose a random number at the beginning of each round. K.Bro wants to know that, for a given sequence, how many rounds are needed to sort this sequence in the best situation. In other words, you should answer the minimal number of rounds needed to sort the sequence into ascending order. To simplify the problem, K.Bro promises that the sequence is a permutation of $1, 2, \dots, N$.

Input

The first line contains only one integer T ($T \leq 200$), which indicates the number of test cases. For each test case, the first line contains an integer N ($1 \leq N \leq 10^6$).

The second line contains N integers a_i ($1 \leq a_i \leq N$), denoting the sequence K.Bro gives you.

The sum of N in all test cases would not exceed 3×10^6 .

Output

For each test case, output a single line 'Case # x : y ', where x is the case number (starting from 1), y is the minimal number of rounds needed to sort the sequence.

Hint:

In the second sample, we choose "5" so that after the first round, sequence becomes "1 2 3 4 5", and the algorithm completes.

Sample Input

```
2
5
5 4 3 2 1
5
5 1 2 3 4
```

Sample Output

```
Case #1: 4
Case #2: 1
```