

The 2016 ACM-ICPC Asia China-Final Contest



Problems

Problem A. Number Theory Problem	1
Problem B. Hemi Palindrome	2
Problem C. Mr. Panda and Strips	3
Problem D. Ice Cream Tower	5
Problem E. Bet	6
Problem F. Mr. Panda and Fantastic Beasts	8
Problem G. Pandaria	9
Problem H. Great Cells	11
Problem I. Cherry Pick	13
Problem J. Mr.Panda and TubeMaster	15
Problem K. Justice Rains From Above	18
Problem L. World Cup	20

Problem A. Number Theory Problem

Input file: Standard Input
Output file: Standard Oupput
Time limit: 1 second

Mr. Panda is one of the top specialists on number theory all over the world.

Now Mr. Panda is investigating the property of the powers of 2. Since 7 is the lucky number of Mr. Panda, he is always interested in the number that is a multiple of 7.

However, we know that there is no power of 2 that is a multiple of 7, but a power of 2 subtracted by one can be a multiple of 7. Mr. Panda wants to know how many positive integers less than 2^N in the form of $2^k - 1$ (k is a positive integer) that is divisible by 7. N is a positive interger given by Mr Panda.

Input

The first line of the input gives the number of test cases, T . T test cases follow.

Each test case contains only one positive interger N .

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 answer.

Limits

- $1 \leq T \leq 100$.
- $1 \leq N \leq 10^5$.

Sample input and output

Sample Input	Sample Output
2	Case #1: 0
1	Case #2: 1
4	

Problem B. Hemi Palindrome

Input file: Standard Input
Output file: Standard Output
Time limit: 1 second

A group of historians sneaked into Ruins of Corvus and found some mysterious binary strings. They worked hard and found that those are all *Hemi Palindrome*. A binary string is called *Hemi Palindrome* if it reads the same forwards as backwards by ignoring digits in odd indices or even indices (1-based index). E.g. “101111” is *Hemi Palindrome* as the numbers on odd indices is “111”. “11001”, “10100” are both *Hemi Palindromes* but “101001”, “11000” are not.

Mr. Panda loves *Hemi Palindromes* so much and he wants to know the K^{th} smallest *Hemi Palindrome* with length N in lexicographic order.

Input

The first line of the input gives the number of test cases, T . T lines follow. Each line contains two integers N and K .

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 result of *Hemi Palindrome* written in binary format. If there are less than K *Hemi Palindromes* with length N , y is “NOT FOUND!”.

Limits

- $1 \leq T \leq 100$.
- $1 \leq N \leq 10^5$.
- $1 \leq K \leq 10^{18}$.

Sample input and output

Sample Input	Sample Output
4	Case #1: 00
2 1	Case #2: 010
3 3	Case #3: 100
3 5	Case #4: NOT FOUND!
3 100	

Problem C. Mr. Panda and Strips

Input file: Standard Input
Output file: Standard Output
Time limit: 3 seconds

Mr. Panda loves colorful strips very much. One day, his friend bought him a long strip with N colors in a row. C_0, C_1, \dots, C_{N-1} from left to right. Unfortunately, Mr. Panda only likes strips with distinct colors, so his friend is going to cut the strip into several parts and then either pick one part or glue **TWO** of them to form a new strip A_0, A_1, \dots, A_{K-1} . In order to make Mr. Panda happy, the new strip should only consist of distinct colors, which means there doesn't exist $0 \leq i < j < K$ that $A_i = A_j$. His friend also wants the strip to be as long as possible. Find out the maximum possible length of the new strip.

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case starts with an integer N , which means the number of colors in the original strip. Then one line consists of N integers C_0, C_1, \dots, C_{N-1} representing the colors.

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 possible length of the new strip.

Limits

- $1 \leq T \leq 20$.
- $1 \leq N \leq 1000$.
- $1 \leq C_i \leq 10^5$.

Sample input and output

Sample Input	Sample Output
3	Case #1: 3
3	Case #2: 5
1 2 3	Case #3: 1
8	
3 1 2 1 6 1 2 5	
3	
1 1 1	

Note

Case #1:

Cut into 2 parts: [1] [2 3], and then glue the two parts together to form a strip: [1 2 3]; or just cut into one part: [1 2 3] then no need to glue at all.

Case #2:

Cut into 3 parts: [3] [1 2] [6 1 2 5], and then glue the first and the third parts to form a strip: [3 6 1 2 5].



Case #3:

Cut into 3 parts: [1] [1] [1], and then can pick one part to form a new strip: [1].

Problem D. Ice Cream Tower

Input file: Standard Input
Output file: Standard Output
Time limit: 6 seconds

Mr. Panda likes ice cream very much especially the ice cream tower. An ice cream tower consists of K ice cream balls stacking up as a tower. In order to make the tower stable, the lower ice cream ball should be at least twice as large as the ball right above it. In other words, if the sizes of the ice cream balls from top to bottom are $A_0, A_1, A_2, \dots, A_{K-1}$, then $A_0 \times 2 \leq A_1, A_1 \times 2 \leq A_2$, etc.

One day Mr. Panda was walking along the street and found a shop selling ice cream balls. There are N ice cream balls on sell and the sizes are $B_0, B_1, B_2, \dots, B_{N-1}$. Mr. Panda was wondering the maximal number of ice cream towers could be made by these balls.

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case starts with a line consisting of 2 integers, N the number of ice cream balls in shop and K the number of balls needed to form an ice cream tower. The next line consists of N integers representing the size of ice cream balls in shop.

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 maximal number of ice cream towers could be made.

Limits

- $1 \leq T \leq 100$.
- $1 \leq N \leq 3 \times 10^5$.
- $1 \leq K \leq 64$.
- $1 \leq B_i \leq 10^{18}$.

Sample input and output

Sample Input	Sample Output
3	Case #1: 2
4 2	Case #2: 2
1 2 3 4	Case #3: 1
6 3	
1 1 2 2 4 4	
6 3	
1 1 2 2 3 4	

Problem E. Bet

Input file: Standard Input
Output file: Standard Output
Time limit: 1 second

The *Codejamon* game is on fire! Fans across the world are predicting and betting on which team will win the game.

A gambling company is providing betting odds for all teams; the odds for the i^{th} team is $A_i:B_i$. For each team, you can bet any positive amount of money, and you do not have to bet the same amount on each team. If the i^{th} team wins, you get your bet on that team back, plus $\frac{B_i}{A_i}$ times your bet on that team.

For example, suppose that there are two teams, with odds of 5:3 and 2:7 and you bet \$20 on the first team and \$10 on the second team. If the first team wins, you will lose your \$10 bet on the second team, but you will receive your \$20 bet back, plus $\frac{3}{5} \times 20 = 12$, so you will have a total of \$32 at the end. If the second team wins, you will lose your \$20 bet on the first team, but you will receive your \$10 bet back, plus $\frac{7}{2} \times 10 = 35$, so you will have a total of \$45 at the end. Either way, you will have more money than you bet (\$20+\$10=\$30).

As a greedy fan, you want to bet on as many teams as possible to make sure that as long as one of them wins, you will always end up with **more** money than you bet. Can you figure out how many teams you can bet on?

Input

The input starts with one line containing exactly one integer T , which is the number of test cases.

Each test case starts with one line containing an integer N : the number of teams in the game. Then, N more lines follow. Each line is a pair of numbers in the form $A_i:B_i$ (that is, a number A_i , followed by a colon, then a number B_i , with no spaces in between), indicating the odds for the i^{th} team.

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 teams that you can bet on, under the conditions specified in the problem statement.

Limits

- $1 \leq T \leq 100$.
- $1 \leq N \leq 100$.
- $0 < A_i, B_i < 100$.
- Both A_i and B_i have **at most** 3 digits after the decimal point.

Sample input and output

Sample Input	Sample Output
1 3 1:1.1 1:0.2 1.5:1.7	Case #1: 2

Note

In sample case #1, one optimal strategy is to bet 1.5 dollars on the first team and 1.5 dollars on the third team. If the first team wins, you will get $1.5 + 1.5 \times (1.1/1) = 3.15$ dollars back, and if the third team wins, you will get $1.5 + (1.7/1.5) \times 1.5 = 3.2$ dollars back. Both of these are higher than the total money that you bet ($1.5 + 1.5 = 3$ dollars).

However, there is no way to bet on all three teams and be guaranteed a profit.

Problem F. Mr. Panda and Fantastic Beasts

Input file: Standard Input
Output file: Standard Output
Time limit: 2 seconds

Mr. Panda loves fantastic beasts, he is going to use a magic spell to figure out where to find them. Given N strings S_1, S_2, \dots, S_N where S_i only contains lowercase alphabet letters, the magic spell is the shortest string which is only contained in the first string as a substring. If there is at least one solution, output the lexicographically smallest one.

If there is no solution, output “Impossible” instead.

Input

The first line of the input gives the number of test cases, T . T test cases follow.

Each test case contains an integer N which is the number of strings.

Following is N lines, each line has a non-empty string S_i which only contains lowercase alphabet letters.

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 the answer string or “Impossible”.

Limits

- $1 \leq T \leq 42$.
- $2 \leq N \leq 50000$.
- $N \leq S_1 + S_2 + \dots + S_N \leq 250000$.
- the sum of S_i in all test cases doesn't exceed 3×10^6 .

Sample input and output

Sample Input	Sample Output
3 2 aba bab 3 qnu cvbb bnu 3 a aa aaa	Case #1: aba Case #2: q Case #3: Impossible

Problem G. Pandaria

Input file: Standard Input
Output file: Standard Output
Time limit: 3 seconds

Mr. Panda has many gardens. For each garden i , there is exactly one flower with color c_i inside. Between gardens, there are many roads connecting them. Each road has a cost related to it.

Mr. Panda likes to walk inside the gardens and pick up as many same color flowers as possible. But sometimes he doesn't want to spend time walking on the roads with cost too high. Here comes the question, every time Mr. Panda tells you which garden he starts to pick flowers and the maximal cost w he can bear (only roads with cost no more than w can be accessible). Please find out which color of flowers Mr. Panda can pick most. If there are more than one color, find the one with smallest color id.

Input

The first line of the input gives the number of test cases, T . T test cases follow.

Each test case begins with 2 integer N and M . N is the number of gardens, M is the number of roads.

Following line has N integers which represents the color of flowers, the i^{th} number is the color of flower on the i^{th} garden.

Following M lines describes roads between garden. Each line has 3 integers x, y, w , representing there is a road with cost w between x^{th} and y^{th} garden.

Then comes a line with an integer Q representing the number of queries. The following Q lines each consists of 2 integers: x, w , representing that Mr. Panda starts at x^{th} garden, and the maximal cost he can bear is w . Assume **last** is the previous query answer, let x and w xor with **last**. For the first query of each test case, **last** = 0.

Output

For each test case, first output one line containing "Case #x:", where **x** is the test case number (starting from 1).

The following Q lines each consists of a integer which is the color id of the maximum same color flowers Mr. Panda can pick most.

Limits

- $1 \leq T \leq 15$.
- $1 \leq N \leq 10^5$.
- $1 \leq M, Q \leq 2 \times 10^5$.
- $1 \leq x, y, c_i \leq N$.
- $1 \leq w \leq 10^6$.
- For 60% of the test cases, $N \leq 5000$ holds.

Sample input and output

Sample Input	Sample Output
1 5 6 2 1 1 3 2 1 2 2 1 3 4 2 3 7 3 4 5 4 5 6 5 3 3 4 1 1 0 0 5 5 6 11	Case #1: 2 1 3 1

Note

The real queries of sample are [(1, 1), (2, 2), (4, 4), (5, 8)].

Problem H. Great Cells

Input file: Standard Input
Output file: Standard Output
Time limit: 2 seconds

Mr. Panda likes playing puzzles with grid paper. Recently he invented a new rule to play with the grid paper.

At the beginning, he grabs a grid paper with N rows and M columns. Then he fills each cell an integer in the range of $[1, K]$. After filling all the cells, he starts finding *Great* cells in the grid. A cell is called *Great* cell if and only if it meets the following 2 conditions:

- The value in the cell is **strictly** larger than other cells in the same row.
- The value in the cell is **strictly** larger than other cells in the same column.

Now Mr. Panda is wondering how many different ways he can fill the grid paper so that there are **exactly** g *Great* cells.

As Mr. Panda likes simple conclusion number, let's just tell him the value

$$\sum_{g=0}^{NM} (g+1) \cdot A_g \pmod{(10^9+7)}$$

A_g represents the number of different ways Mr. Panda can fill the grid paper such that there are exactly g *Great* cells.

Input

The first line of the input gives the number of test cases, T . T lines follow.

Each line represents a test case containing 3 integers N , M representing the number of rows and columns of the grid paper, K representing the range for filling numbers.

Output

For each test case, first output one line containing “Case #x: y”, where x is the test case number (starting from 1), y is the simple conclusion number for Mr. Panda.

Limits

- $1 \leq T \leq 20$.
- $1 \leq N, M, K \leq 200$.

Sample input and output

Sample Input	Sample Output
3	Case #1: 24
2 2 2	Case #2: 88
2 3 2	Case #3: 487890625
3 4 5	

Note

For the first sample, $A_0 = 10, A_1 = 4, A_2 = 2, A_3 = A_4 = 0$, thus the answer is $10+4\times 2+2\times 3 = 24$.

Problem I. Cherry Pick

Input file: Standard Input
Output file: Standard Output
Time limit: 10 seconds

Jane is visiting a cherry orchard which consists of N cherry trees. The orchardman tells Jane that she can pick as many cherries as she want and pay at the checkout counter. As Jane buys cherries directly from the orchardman, the price is really cheap: one local currency per cherry! There are M different types of banknotes in her wallet. The denomination of the i^{th} type is C_i , and there are infinite banknotes for each type.

Jane randomly picks cherries from cherry trees. For each tree, she either picks one cherry from it or skip that tree. She picks a cherry from each cherry tree with probability $P\%$. The picking choice on each cherry tree is independent.

To make business smooth, the orchardman never gives changes. Jane always pays the minimal amount of money for the cherries she picks. For example, suppose that Jane only has one type of banknote with denomination of 10, she will pay 10 local currency for 9 cherries, 10 local currency for 10 cherries, and 20 local currency for 11 cherries.

Being a shrewd girl, she wonders how much extra money will be paid because the orchardman never gives changes. In the above example, she will pay extra 1 local currency when picking 9 cherries; no extra money will be paid when picking 10 cherries; and extra 9 local currency when picking 11 cherries.

She turns to you, the loyal and clever steward, to calculate the expectation of the extra money she will pay. It can be proved that the expectation multiplies 100^N is always an integer. To make life easier, you are only required to calculate the expectation multiplies 100^N and then mod $10^9 + 7$.

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case starts with three integers N , M and P , indicating the number of cherry trees in the orchard, the number of different types of banknotes in Janes wallet, and the probability that Jane picks a cherry from each cherry tree. The following line contains M integers C_1, C_2, \dots, C_M , describing the denomination of each type of banknotes.

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 result.

Limits

- $1 \leq T \leq 20$.
- $1 \leq N \leq 10^9$.
- $1 \leq M \leq 100$.
- $0 \leq P \leq 100$.
- $1 \leq C_i \leq 10000$.

- $1 \leq C_i \times C_j \leq 10000$, for all $i \neq j$.
- all C_i are distinct.
- For 25% of the test cases, $M = 1$ holds.

Sample input and output

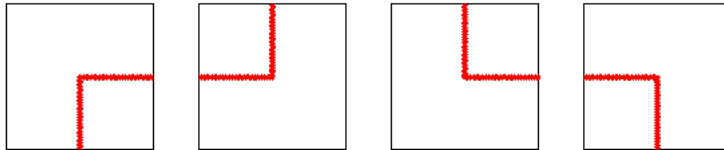
Sample Input	Sample Output
3 2 1 50 3 2 3 100 3 4 5 2 2 50 1 3	Case #1: 12500 Case #2: 10000 Case #3: 0

Problem J. Mr.Panda and TubeMaster

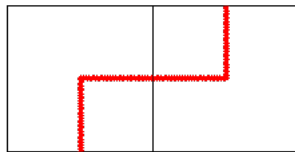
Input file: Standard Input
Output file: Standard Ouput
Time limit: 5 seconds

Mr.Panda loves playing games. Recently he falls in love with a game named TubeMaster.

In TubeMaster, players can build tubes in cells in a $N \times M$ table. Each cell can be either left empty or put in exact one of the following four kinds of tubes.



When two adjacent cells sharing a same edge are connected by tubes (e.g. the below picture), the player will gain some score (the score will be described in the Input section).



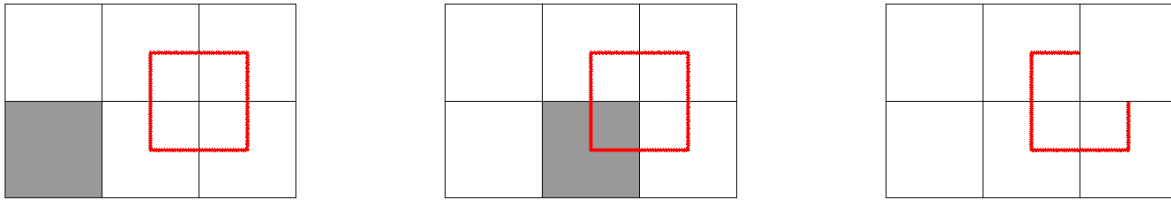
Some of the cells are considered as essential cells, which means each of such cells is required to contain a tube, otherwise the player loses the game.

Before the end of the game, the table needs to be a valid configuration, otherwise the player loses the game.

A valid configuration has to meet the following conditions:

- Each cell needs to contain either no tube or a tube which is in a tube cycle.
- Each of the essential cells needs to contain a tube.

Note that multiple tube cycles might appear in a valid configuration simultaneously and an empty table can also be a valid configuration when there are no essential cells.



In the tables above, the left one is a valid configuration while the middle one and the right one are not. (grey cells are essential cells)

Mr.Panda wants to win the game with a maximum accumulation of scores. Could you please help him calculate the value?

Input

The first line of the input gives the number of test cases, T .

T test cases follow. Each test case starts with a line containing two numbers separated by a space, N and M indicating the number of rows and the number of columns of the table.

Then N lines follow, each line contains $M - 1$ numbers separated by spaces where $\text{ScoreC}_{i,j}$ (i.e. the j^{th} number in the i^{th} line) indicates the score of connecting the cell (i, j) and cell $(i, j + 1)$ with tubes.

Then $N - 1$ lines follow, each line contains M numbers separated by spaces where $\text{ScoreR}_{i,j}$ (i.e. the j^{th} number in the i^{th} line) indicates the score of connecting the cell (i, j) and cell $(i + 1, j)$ with tubes.

Then there will be a line containing a number, E the number of essential cells.

Finally E lines follow, each line contains two number separated by a space, R_i and C_i indicating that the cell (R_i, C_i) is an essential cell.

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 accumulation of scores if Mr.Panda wins the game. If Mr.Panda cannot win the game, output “Impossible” for y.

Limits

- $1 \leq T \leq 100$.
- $1 \leq N, M \leq 30$.
- $-500 \leq \text{ScoreC}_{i,j}, \text{ScoreR}_{i,j} \leq 500$.
- $0 \leq E \leq 100$.
- $1 \leq R_i \leq N$.
- $1 \leq C_i \leq M$.
- For 40% of the test cases, $N, M \leq 10$ holds.

Sample input and output

Sample Input	Sample Output
2 4 4 0 0 -1 0 1 0 0 -1 -1 0 1 0 1 0 1 0 -1 -1 0 0 1 1 -1 -1 1 3 3 2 3 0 0 0 0 0 0 0 2 1 1 2 3	Case #1: 2 Case #2: Impossible

Problem K. Justice Rains From Above

Input file: Standard Input
Output file: Standard Output
Time limit: 10 seconds

If you have ever played the game Overwatch, you may know the hero Pharah, an offense hero that is capable of flight.

Pharah has a powerful ultimate ability called "Barrage" which is one of the most devastating abilities in the game. By casting it, Pharah will direct a continuous salvo of mini-rockets to destroy groups of enemies.

Let's look at how it works. The map of this game can be regarded as a three-dimensional space, all players are points in this space. For Barrage, the rockets cover a large area which contains all points where the angle between each point's direction to Pharah and the direction of Pharah's crosshair is not larger than α . And Pharah must remain stationary while the ability is active, include her crosshair.



Now you have come to the map Numbani and chosen Pharah as your hero. There are N enemies on the ground (i.e. $z = 0$), with unique positions. By using Pharah's ability "Jump Jet", you have flown to a vantage point (i.e. $z > 0$), and the ultimate ability is ready. If you want to make the "Play of the Game", you must kill as many enemies as possible.

Here comes a "Simple Geometry" problem, what's the largest number of enemies can be killed by using your ultimate ability if you choose the best crosshair direction?

Input

The first line of the input gives the number of test cases, T . T test cases follow.

Each test case begins with two integers N and α , the number of enemies and the angle range of Barrage.

Next line has three integers x , y and z , indicating the coordinates of Pharah's position.

Next N lines, each line has two integers x , y , indicating the coordinates of each enemy's position on the ground.

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 largest number of enemies that can be killed by casting Barrage.

Limits

- $1 \leq T \leq 100$.
- $1 \leq N \leq 1000$.
- $0 < \alpha < 90$.
- $|x|, |y| \leq 1000$.
- $0 < z \leq 1000$.
- For 90% of the test cases, $N \leq 6$ holds.

Sample input and output

Sample Input	Sample Output
2 2 45 0 0 3 1 -2 -5 2 4 89 1 1 1 30 0 0 50 -70 0 0 -90	Case #1: 2 Case #2: 3

Problem L. World Cup

Input file: Standard Input
Output file: Standard Output
Time limit: 1 second

Here is World Cup again, the top 32 teams come together to fight for the World Champion.

The teams are assigned into 8 groups, with 4 teams in each group. Every two teams in the same group will play a game (so there are totally 6 games in each group), and the winner of this game gets **3** points, loser gets **0** point. If it is a tie game, both teams get **1** point.

After all games finished, we get the scoreboard, but we forget the result of each game, can you help us to figure the result of each game? We only care about the win/lose/tie result of each game, but we don't care the goals in each game.

Input

The input starts with one line containing exactly one integer T , which is the number of test cases. Each test case contains four space-separated integers A, B, C, D , in a line, which indicate the points each team gets after all 6 games.

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 "Yes" if you can point out the result of each game, or "No" if there are multiple game results satisfy the scoreboard, or "Wrong Scoreboard" if there is no game result matches the scoreboard.

Limits

- $1 \leq T \leq 100$.
- $0 \leq A, B, C, D \leq 100$.

Sample input and output

Sample Input	Sample Output
3	Case #1: Yes
9 6 3 0	Case #2: No
6 6 6 0	Case #3: Wrong Scoreboard
10 6 3 0	

Note

In **sample case #1**, the only scenario will be: the first team wins all the three games it plays, the second team loses to the first team and wins the other two, the third team only wins the game with the fourth, and the fourth team lose all the games.

In **sample case #2**, the fourth team loses all the games, and the first three teams get into a winning-cycle, but there may be two different winning-cycles: first team wins second team, second team wins third team, third team wins first team OR first team wins third team, third team wins second team, second team wins first team. We can't figure which winning-cycle is the actual game result.



In **sample case #3**, the first team get 10 points, but no team could get more than 9 points by play three games, so it is a wrong scoreboard.