Problem A. Dogs and Cages

Jerry likes dogs. He has N dogs numbered $0, 1, \ldots, N-1$. He also has N cages numbered $0, 1, \ldots, N-1$. Everyday he takes all his dogs out and walks them outside. When he is back home, as dogs can't recognize the numbers, each dog just randomly selects a cage and enters it. Each cage can hold only one dog.

One day, Jerry noticed that some dogs were in the cage with the same number of themselves while others were not. Jerry would like to know what's the expected number of dogs that are **NOT** in the cage with the same number of themselves.

Input

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

Each test case contains only one number N, indicating the number of dogs and cages.

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 expected number of dogs that are **NOT** in the cage with the same number of itself.

`y` will be considered correct if it is within an absolute or relative error of 10^{-6} of the correct answer.

Limits

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

Example

standard input	standard output
2 1 2	Case #1: 0.0000000000 Case #2: 1.0000000000

Note

In the first test case, the only dog will enter the only cage. So the answer is 0.

In the second test case, if the first dog enters the cage of the same number, both dogs are in the cage of the same number, the number of mismatch is 0. If both dogs are not in the cage with the same number of itself, the number of mismatch is 2. So the expected number is $\frac{0+2}{2} = 1$.

Problem B. Same Digit

Little Mono is a smart child, he can do complex arithmetical operations quickly. But he only knows one digit $D(1 \leq D \leq 9)$. He would like to use the only digit he knows to make expressions to represent integer numbers.

A valid expression can be generated like this:

1. Any number consists of only digit D are valid expressions. E.g. if $D = 1$, then 1, 11, 111, ... are all valid expressions.
2. If A and B are valid expressions, then $(A) + (B)$ is a valid expression.
3. If A and B are valid expressions, then $(A) - (B)$ is a valid expression.
4. If A and B are valid expressions, then $(A) * (B)$ is a valid expression.
5. If A and B are valid expressions, then $(A)/(B)$ is a valid expression. (/ here produces exact value, not integer division and expression A must produce a non-zero value)
6. If A and B are valid expressions, then $(A)^{(B)}$ is a valid expression. (expression A must produce a non-negative value)
7. If A is valid expression, then $\sqrt{(A)}$ is a valid expression.(expression A must produce a non-negative value) **IMPORTANT**: a valid expression can contain at most ten $\sqrt{(A)}$ operations.
8. If A is valid expression, then $(A)!$ is a valid expression.(! here means factorial, and expression A must produce a non-negative integer)
9. If A is valid expression, then $-(A)$ is a valid expression.($-$ here means negative)

Now Little Mono would like to know the minimal number of Ds he needs to use in order to represent integer N.

Input

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

Each test case contains one line consists of 2 integers D, N, indicating the digit Little Mono knows and the integer Little Mono would like to represent.

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 minimal number of Ds Little Mono has to use.

Limits

- $1 \leq T \leq 900$.
- $1 \leq D \leq 9$.
- $1 \leq N \leq 100$.

Example

standard input	standard output
3 1 10 4 64 8 50	Case #1: 3 Case #2: 2 Case #3: 5

Note

In the first test case, Little Mono can use 3 1s as $11 - 1$ to represent 10.

In the second test case, Little Mono can use 2 4s as $\sqrt{\sqrt{\sqrt{4}}}^{(4!)}$ to represent 64.

Problem C. Rich Game

One day, the god of sheep wanted to play badminton but he couldn't find an opponent. So he asked Mr. Panda to play with him. He said: "Each time I win one point, I'll pay you X dollars. For each time I lose one point, you should give me Y dollars back."

The god of sheep was going to play K sets of badminton games. For each set, anyone who wins at least 11 points **and** leads by at least 2 points will win this set. In other words, normally anyone who wins 11 points first will win the set. In case of **deuce** (E.g. 10 points to 10 points), it's necessary to lead by 2 points to win the set.

Mr. Panda is really good at badminton so he could make each point win or lose at his will. But he had no money at the beginning. He needed to earn money by losing points and using the money to win points. Mr. Panda couldn't owe the god of sheep money as god never lends money. What's the maximal number of sets can Mr. Panda win if he plays cleverly?

Input

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

Each test case contains 3 numbers in one line, X, Y, K, the number of dollars earned by losing a point, the number of dollars paid by winning a point, the number of sets the god of sheep is going to play.

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 sets Mr. Panda could win.

Limits

- $1 \leq T \leq 10^5$.
- $1 \leq X, Y, K \leq 1000$.

Example

standard input	standard output
2 10 10 1 10 10 2	Case #1: 0 Case #2: 1

Note

In the first test case, Mr. Panda don't have enough money to win the only set, so he must lose the only set.

In the second test case, Mr. Panda can lose the first set by 0:11 and he can earn 110 dollars. Then win the second set by 11:0 and spend 110 dollars.

Problem D. Mr. Panda and Circles

Mr. Panda likes creating and solving mathematical puzzles. One day, Mr. Panda came up with a puzzle while he was playing the following game with Mrs. Panda:

In a plane, there are M points $(0,0),(1,0),..,(M-2,0),(M-1,0)$ in a segment. You are also given N circles, the radius of i^{th} circle is R_i. In the game, you are allowed to put center of any circle into one of the M points without making circles overlap (that is, if the intersection of two circles has a positive area), and **ALL** circles should be used.

An arrangement of circles is considered as valid if every circle's center is in one of the M points. Mr. Panda wanted to know length of empty units which are not covered by any circle in the segment from $(0,0)$ to $(M-1,0)$.

Because there are too many arrangements, Mr. Panda only wanted to know $\sum L_i^2$ modulo $1,000,000,007$ where L_i is length of empty units in the i^{th} arrangement.

The puzzle has confused Mr. Panda for a long time. Luckily, Mr. Panda knows you are in this contest. Could you help Mr. Panda to solve the puzzle?

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 two integers - N, the number of circles, and M, the number of points.

Then, a line consisting of N integer numbers follows, the i^{th} number R_i indicates radius of the i^{th} circle.

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 number that Mr. Panda wants to know for the x^{th} input data set.

Limits

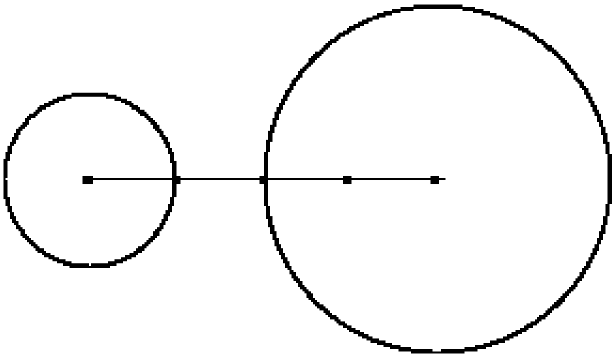
- $1 \le T \le 30$.
- $1 \le N \le 10^5$.
- $2 \le M \le 10^{18}$.
- $1 \le R_i \le 10^5$.

Example

standard input	standard output
5 3 6 1 1 1 2 5 1 2 2 6 1 2 3 2 1 1 1 1 10 50	Case #1: 12 Case #2: 2 Case #3: 14 Case #4: 0 Case #5: 0

Note

In the second sample case, an arrangement with an empty unit is:



The other arrangement with an empty unit can be produced by mirroring the arrangement above. It is obvious that there is no arrangement with two or more empty units. Thus the answer is $1^2 + 1^2 = 2$.

Problem E. Evil Forest

The Bear, king of the forest, loves painting very much. His masterpiece – "Back in time" is very famous around all over the forest. Recently he wants to hold a series painting competition to select those animals who are good at painting as well. So the Bear appoints his minister Tiger to help him prepare the painting competitions.

Tiger owns a stationery factory which produces sketchpad, so he wants to let all the competitions use the sketchpads produced by his factory privately. The Bear plans to hold N painting competitions, and informs Tiger of the number of competitors who are ready for each competition. One competitor needs only one sketchpad in one competition. For each competition, at least 10% extra sketchpads are required as backup just in case.

The Tiger assigns you, his loyal subordinate to calculate the minimum number of sketchpads produced by his factory.

Input

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

For each test case, the first line contains an integer N, where N is the number of competitions. The next line contains N integers $a_1, a_2, \ldots, a_N$ representing the number of competitors in each competition.

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 minimum number of sketchpads that should be produced by Tiger's factory.

Limits

- $1 \le T \le 100$.
- $1 \le N \le 1000$.
- $1 \le a_i \le 100$.

Example

standard input	standard output
1 6 13 11 11 11 13 11	Case #1: 82

Note

For the first test case, two more sketchpads should be prepared for each painting competition.

Problem F. Fair Lottery

There is a lottery going to happen among N groups of people. Group i consists of a_i persons. In the end, at most M persons will be the winner of this lottery.

There is one extra restriction for this lottery: for each group, either all or none of the person in this group are winners. In other words, no partial winners allowed for each group.

A lottery program is defined as a set of outcomes. The outcome announces the winners of the lottery. Each outcome has it's own probability to happen, and the sum of probabilities among all outcomes equals to 1.0.

A fair lottery program is a lottery program that makes each person having the same probability p of winning the lottery. Find the maximum probability p over all fair lottery programs.

For example, suppose there are 3 groups: A, B, and C, each consists of 1, 2, and 2 persons. At most 3 persons will be the winner of the lottery.

One possible fair lottery program is: it consists of only one outcome that no one wins. So each person has the same probability of 0% winning the lottery.

A better fair lottery program is: there are two outcomes, each happening with probability of 50%. Outcome 1 announces person in group A and B as winners; and outcome 2 announces person in group C as winners. Hence each person having the winning probability of 50%. It's easy to verify that 50% is the maximum probability over all lottery programs.

Input

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

Each test case contains 2 lines, the first line consists of 2 numbers N, M indicating the number of groups and the max number of lottery winners.

The second line consists of N numbers $a_1, a_2, \ldots, a_N$ indicating the number of persons in each group.

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 probability p for each one to be the winner over all fair lottery programs.

`y` will be considered correct if it is within an absolute or relative error of 10^{-6} of the correct answer.

Limits

- $1 \le T \le 100$.
- $1 \le N \le 10$.
- $1 \le M \le 100$.
- $1 \le a_i \le 100$.

Example

standard input	standard output
2 3 3 1 2 2 4 2 1 1 1 2	Case #1: 0.5000000000 Case #2: 0.4000000000

Note

The first test case is the sample in the problem.

In the second test case, the best lottery program is: Assume groups are A(1), B(1), C(1), D(2), For $[0, 0.2)$ persons in A, B are the winners. For $[0.2, 0.4)$ persons in B, C are the winners. For $[0.4, 0.6)$ persons in A, C are the winners. For $[0.6, 1)$ persons in D are the winners.

Problem G. Alice's Stamps

Alice likes to collect stamps. She is now at the post office buying some new stamps.

There are N different kinds of stamps that exist in the world; they are numbered 1 through N. However, stamps are not sold individually; they must be purchased in sets. There are M different stamp sets available; the i^{th} set contains the stamps numbered L_i through R_i. The same stamp might appear in more than one set, and it is possible that one or more stamps are not available in any of the sets.

All of the sets cost the same amount; because Alice has a limited budget, she can buy at most K different sets. What is the maximum number of different kinds of stamps that Alice can get?

Input

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

Each test case begins with a line containing three integers: N, M, and K: the number of different kinds of stamps available, the number of stamp sets available, and the maximum number of stamp sets that Alice can buy.

M lines follow; the i^{th} of these lines represents the i^{th} stamp set and contains two integers, L_i and R_i, which represent the inclusive range of the numbers of the stamps available in that set.

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 different kinds of stamp that Alice could get.

Limits

- $1 \le T \le 100$.
- $1 \le K \le M$.
- $1 \le N, M \le 2000$.
- $1 \le L_i \le R_i \le N$.

Example

standard input	standard output
2 5 3 2 3 4 1 1 1 3 100 2 1 1 50 90 100	Case #1: 4 Case #2: 50

Note

In sample case #1, Alice could buy the first and the third stamp sets, which contain the first four kinds of stamp. Note that she gets two copies of stamp 3, but only the number of different kinds of stamps matters, not the number of stamps of each kind.

In sample case #2, Alice could buy the first stamp set, which contains 50 different kinds of stamps.

Problem H. Equidistance

Given M points in Euclidean space of dimension N, where the distances between any pair of points is 1.0, how many points can be added that the equidistance restriction still holds?

Print the maximum number of points can be added, and the coordinates of the points. Recall that the distance of two points $(a_1, a_2, \ldots, a_N)$ and $(b_1, b_2, \ldots, b_N)$ in N dimensional Euclidean space is defined as:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_N - b_N)^2}$$

Input

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

The first line of each test case contains two integer N and M, indicating the dimension of the Euclidean space and the number of given points. Then follows M lines, each line contains N real numbers indicating the coordinates of the given points. It is guaranteed that the distances between any two given points is 1.0.

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 points can be added that the equidistance restriction still holds.

The following `y` lines each contains N real numbers specifying the coordinates of the added points. There could be multiple answers to the given input. The answer is accepted as long as for each pair of points, the absolute difference between the distance and 1.0 is smaller than 10^{-8}.

Limits

- $1 \le T \le 100$.
- $1 \le N \le 100$.
- $1 \le M$.
- The coordinates' absolute value of given points will be strictly less than 100.

Example

standard input	standard output
2 1 1 0.0000000000 2 2 1.0000000000 0.0000000000 2.0000000000 0.0000000000	Case #1: 1 1.0000000000 Case #2: 1 1.5000000000 0.8660254038

Problem I. Inkopolis

Inkopolis is the city in which Inklings live in, it can be regarded as an undirected connected graph with exactly N vertices and N edges between vertices. It is guaranteed that the graph doesn't contain duplicated edges or self loops. Inklings can splatter a special type of colored ink to decorate the roads they live in. Inklings are capricious so they often change the color of the roads to celebrate the upcoming Splatfest.

The Splatfest lasts for exactly M days, on each day Inklings will splatter ink on exactly one road, the color on this road will be covered by the new color of the ink. At the end of each day, they wonder how many different colored regions are there in the Inkopolis. A colored region is a set of connected roads with same color, to be clear, two roads are in the same colored region if they have the same color and share a common vertex.

Input

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

For each test case, the first line contains two integers N and M, where N is the number of vertexes and roads in Inkopolis, and M is the number of days that Splatfest lasts.

Following N lines describe the roads between the vertexes. Each line contains 3 integers x, y, c, representing there is a road with color c between the x^{th} and the y^{th} vertex.

Next following M lines describe the operations on each day. Each line contains 3 integers x, y, c, representing an operation that Inklings splatter the ink with color c to the road between the x^{th} and the y^{th} vertex, it is guaranteed that there is such a road.

Output

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

The following M lines each consists of an integer which is the number of colored regional in Inkopolis after each day.

Limits

- $1 \le T \le 100$.
- $3 \le N \le 2 \times 10^5$.
- $1 \le M \le 2 \times 10^5$.
- $1 \le x, y, c \le n$.
- $\sum N, \sum M \le 10^6$.

Example

standard input	standard output
2 4 3 4 2 4 2 3 3 3 4 2 1 4 1 3 4 2 2 3 4 3 4 3 4 4 1 2 1 2 3 1 3 4 1 4 1 1 1 2 2 3 4 2 2 3 2 4 1 4	Case #1: 4 3 3 Case #2: 2 4 2 2

Problem J. Subway Chasing

Mr. Panda and God Sheep are roommates and working in the same company. They always take subway to work together. There are N subway stations on their route, numbered from 1 to N. Station 1 is their home and station N is the company.

One day, Mr. Panda was getting up later. When he came to the station, God Sheep has departed X minutes ago. Mr. Panda was very hurried, so he started to chat with God Sheep to communicate their positions. The content is when Mr. Panda is between station A and B, God Sheep is just between station C and D.

B is equal to $A+1$ which means Mr. Panda is between station A and $A+1$ exclusive, or B is equal to A which means Mr. Panda is exactly on station A. Vice versa for C and D. What's more, the communication can only be made no earlier than Mr. Panda departed and no later than God Sheep arrived.

After arriving at the company, Mr. Panda want to know how many minutes between each adjacent subway stations. Please note that the stop time at each station was ignored.

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 consists of 3 integers N, M and X, indicating the number of stations, the number of chat contents and the minute interval between Mr. Panda and God Sheep.

Then M lines follow, each consists of 4 integers A, B, C, D, indicating each chat content.

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 minutes between stations in format $t_1, t_2, \ldots, t_{N-1}$. t_i represents the minutes between station i and $i+1$. If there are multiple solutions, output a solution that meets $0 < t_i \le 2 \times 10^9$. If there is no solution, output "`IMPOSSIBLE`" instead.

Limits

- $1 \le T \le 30$.
- $1 \le N, M \le 2000$.
- $1 \le X \le 10^9$.
- $1 \le A, B, C, D \le N$.
- $A \le B \le A+1$.
- $C \le D \le C+1$.

Example

standard input	standard output
2 4 3 2 1 1 2 3 2 3 2 3 2 3 3 4 4 2 2 1 2 3 4 2 3 2 3	Case #1: 1 3 1 Case #2: IMPOSSIBLE

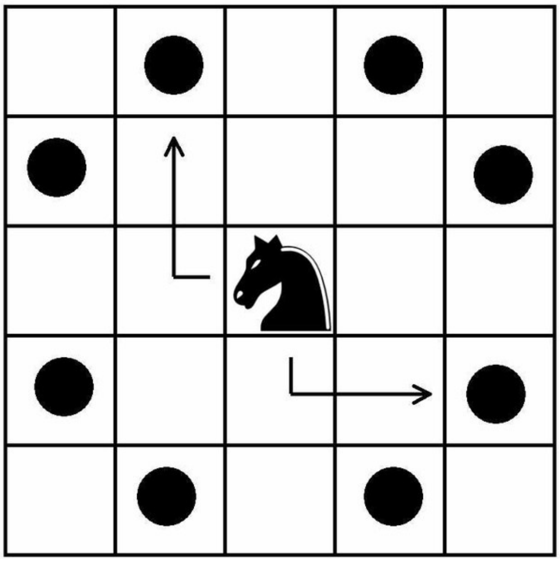
Note

In the second test case, when God Sheep passed the third station, Mr. Panda hadn't arrived the second station. They can not between the second station and the third station at the same time.

Problem K. Knightmare

A knight jumps around an infinite chessboard. The chessboard is an unexplored territory. In the spirit of explorers, whoever stands on a square for the first time claims the ownership of this square. The knight initially owns the square he stands, and jumps N times before he gets bored.

Recall that a knight can jump in 8 directions. Each direction consists of two squares forward and then one square sideways.



After N jumps, how many squares can possibly be claimed as territory of the knight? As N can be really large, this becomes a nightmare to the knight who is not very good at math. Can you help to answer this question?

Input

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

Each test case contains only one number N, indicating how many times the knight jumps.

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 number of squares that can possibly be claimed by the knight.

Limits

- $1 \le T \le 10^5$.
- $0 \le N \le 10^9$.

Example

standard input	standard output
3 0 1 7	Case #1: 1 Case #2: 9 Case #3: 649