Problem A – Aliens attack

The Earth is being attacked by aliens. All nations around the world join together in an alliance group against aliens. The first thing to do is to determine the effect of bomb attacks from aliens on earth. There are total N bomb attacks. There is a special feature that each bomb attack will damage a triangle area on earth.

The alliance needs to know the total area on earth where is damaged by N bomb attacks from the aliens.

Input

The first line contains integer T, which is the number of test cases. Each test case will be described as follows:

- The first line contains number N ($N \le 10$), which is the number of bomb attacks from the aliens.
- Each of the next N line contains information about one bomb attacks, which includes coordinates of the damaged triangle area x1, y1, x2, y2, x3, y3, respectively. ($|x1|, |y1|, |x2|, |y2|, |x3|, |y3| \le 1000$).

Output

Print case number and result for each test case in one line, with exactly 4 digits after decimal point.

Sample

Sample input	Output for sample input
2 -3 -3 -1 3 3 3 -1 -2 -2 -2 3 -1	Case #1: 12.5000

Problem B – Fight back the aliens

To fight against the attacks from the aliens, K most talented men and women are selected to join a special force. They gather in a secret location to train and prepare for the fight back. They plan to have an intensive training for N hours to improve their power. After the i^{th} hour of training, the power of the group will increase by i^K. Thus, the accumulated power S of the whole team after the intensive training is equal to $1^K + 2^K + ... + N^K$. Your task is to calculate S.

Input

The first line contains a number T – the number of test cases. Each of the next T lines contains a test case consisting of 2 integers N ($1 \le N \le 10^9$) and K ($0 \le K \le 40$).

Output

For each test case, print the result modulo 10^9+7 in one line.

Sample

Sample input	Output for sample input
1 2 3	9

Problem C – Tropical biathlon

Biathlon is an athletic contest combining two events. Traditionally, it is an individual event of skiing and rifle shooting that contestants ski around a cross-country trail system, which the total distance is broken up by either two or four shooting rounds, half in prone position, the other half standing. Depending on the shooting performance, extra distance or time are added to contestants' total running distance/time.

It is almost impossible to have such event in a tropical country like Vietnam so we invented the tropical biathlon consisting of cross-country cycling and cross-country running. One more different thing is contestants compete in pairs. The event is hosted in the Northern mountainous area with small dirt roads going through villages.

At the starting point, one contestant has to compete in cycling, while the other has to compete in running. At the some point during the event, the first contestant has to stop cycling, leave the bicycle in a village and start running toward the finish line; the second player needs to get to that village, take the bicycle and cycle to the finish line. The two players might not take the same route but the switch is mandatory and must be done at exactly once (not at the start or finish). The result for a team is the time for both 2 players to finish.

This contest is not only about physical strength, but it is also about planning. Bi and Bo are 2 first time players. During practice, their performance on both cycling and running was very good and almost the same. With the map of the area, they are planning on the switching strategy to get the best result.

The area has N villages, with the starting point at village 1 and the finish line at village N. There are M bidirectional roads. Road i^{th} connects village a_i and b_i. Bi and Bo estimated that for this road, the cycle time is c_i and the running time is r_i. Given all these data, can you help them to calculate the best possible result?

Input

The first line of the input contains a number T - the number of test cases ($T \le 25$). Each test case is described as follows:

- The first line of each test consists of 2 integers N and M ($3 \le N \le 25000$, $2 \le M \le 250000$).
- The next M lines describe M roads. Each consists of 4 integers a_i, b_i, c_i, r_i ($1 \le c_i, r_i \le 2500$).

Output

For each test case, print the best possible result in a single line. The input data guarantees that there will be at least one possible solution.

Sample

Sample input	Output for sample input
2 3 2 1 2 1 2 2 3 1 2 3 2 1 2 3 10 2 3 4 8	3 14

Problem D – Warehouse patrol

ADR is a new e-commerce startup with a big ambition of dominating the e-commerce market in Vietnam. Backed by strong investment, they have just built a giant warehouse in order to improve customer experience.

They have to purchase 2 patrolling robots because of giant size of their warehouse. These robots can accept instruction to travel around the warehouse. The instruction has the format: "x y w h" which means:

1. Starting position is (x, y);
2. Go w steps to the east to position (x + w, y);
3. Go h steps to the north to position (x + w, y + h);
4. Go w steps to the west to position (x, y + h);
5. Go h steps to the south to position (x, y);
6. Repeat steps 2, 3, 4 and 5.

The company wants to set up their instruction and start the 2 robots at the same time. Each robot can move 1 unit in a step. Because these robots do not have automatic anti-collision feature, they have to make sure that the input they pass to the 2 robots does not cause them to collide with each others. Given this data, your task is to determine if the data are safe or not?

Input

The first line of the input contains the number of test cases - T ($T \le 1000$). Each test case consists of 2 lines describing the setup instructions of the 2 robots, each consists of 4 integers x y w h ($0 < x, y, w, h < 10^5$). It is guaranteed that the initial positions of 2 robots are different.

Output

For each test case, print out "safe" if the instructions never cause them colliding, or "collide" if robots will collide with each others.

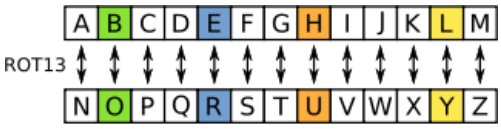
Sample

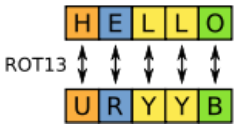
Sample input	Output for sample input
2 1 1 2 2 2 2 2 2 1 1 2 2 3 3 1 1	safe collide

Problem E – ROT-13

ROT13 is a simple letter substitution cipher that replaces a letter with the letter 13 letters after it in the alphabet. Because there are 26 letters (2×13) in the basic Latin alphabet, ROT13 is its own inverse; that is, to undo ROT13, the same algorithm is applied, so the same action can be used for encoding and decoding.

Applying ROT13 to a piece of text merely requires examining its alphabetic characters and replacing each one by the letter 13 places further along in the alphabet, wrapping back to the beginning if necessary. A becomes N, B becomes O, and so on up to M, which becomes Z, then the sequence continues at the beginning of the alphabet: N becomes A, O becomes B, and so on to Z, which becomes M. The lower case characters follow the same rules: a becomes n, b becomes o, … Only those letters which occur in the English alphabet are affected; numbers, symbols, whitespace, and all other characters are left unchanged.





Given a string S consisting of characters with ASCII code larger than 32 (length(S) <= 1000), your task is to calculate ROT13(S).

Input

The first line of the input contains the number of test cases – T. Each test case is printed in one line.

Output

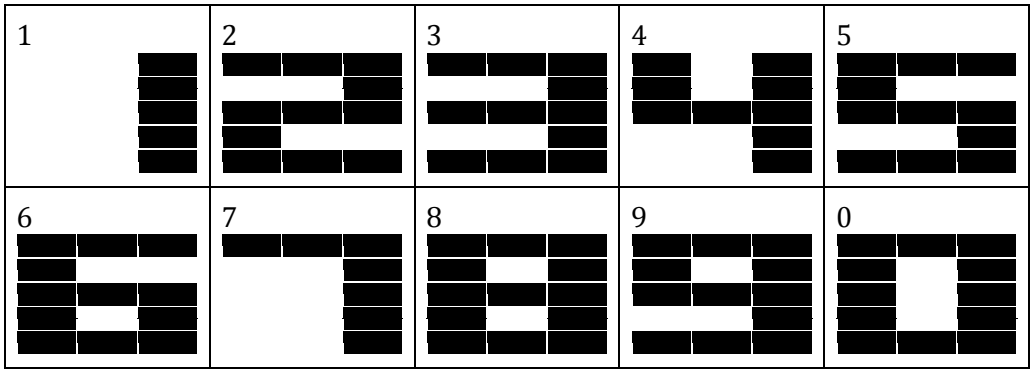
For each test case in the input, print the string result after applying ROT13.

Sample

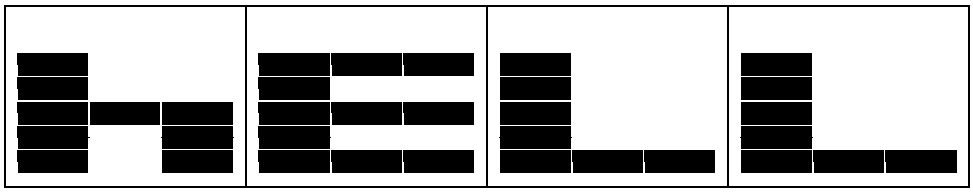
Sample input	Output for sample input
2 The_Quick_Brown_Fox_Jumps_Over_The_Lazy_Dog Gur_Dhvpx_Oebja_Sbk_Whzcf_Bire_Gur_Ynml_Qbt	Gur_Dhvpx_Oebja_Sbk_Whzcf_Bire_Gur_Ynml_Qbt The_Quick_Brown_Fox_Jumps_Over_The_Lazy_Dog

Problem F – Countdown clock

To prepare for the next ACM/ICPC regional contest, a countdown clock has been built to count how many days left to the contest. The clock consists of 9 blocks of light bulbs in a row. Each block is a set of 15 LED light bulbs arranged in the form of 5 rows by 3 columns to represent one digit. All the gaps between 2 consecutive blocks have the same size. The digits are represented as followed image:



However, the careless construction workers put the countdown clock upside down. So the number does not appear correctly as it should be and sometimes, a look-alike meaningful word can appear like “hELL” (7734).



Long realized the problem, and the current count today is N. He is wondering, in the next D days, without putting back the clock in the correct way, how many times the clock will actually show the “correct-look” of the countdown. Assuming that, the clock will never show leading zeros. Thus “hELLO” - “07734” never appears, but “hELL” - “7734” does. The correct-look can be defined as:

- For every position, the digit of the upside down number must look equal to the correct number.
- Spacing between every 2 consecutive digits must be similar. Thus, the upside down of “101” does not look like “101”, although the 1 and 1-upside down have the correct-look.

Input

The first line contains the number of test cases - T ($T \le 10000$). Each test case is written in a single line consisting of 2 integers N and D. ($0 \le D \le N < 10^9$).

Output

For each test case, print the number of times the clock looks correct in one line.

Sample

Sample input	Output for sample input
4 9 2 88 1 102 2 1 1	1 0 0 1

Explanation:
Case 1: In the next 2 days, 8 and 7 will show up. Only 8 is correct.
Case 2: We do not count today.
Case 3: As the previous explanation, the upside down of 101 does not look like 101.
Case 4: Upside down of 0 is 0.

Problem G – Googol-th Independence Day

Country Alpha is about to celebrate their googol-th Independence Day. The government decided that they will hold a big parade event.

This country has N cities numbered 1 to N, the capital city is city 1. These cities are connected by M one-way roads. In the event, there will be N-1 parade groups. The u^{th} group will start at the capital, go by some (or no) cities, and end at city (u + 1). In order to make a memorable event, they decided that all roads, which the parade travelled by, will be decorated. However, the government also wants to minimize the decoration cost. Given the cost to decorate each road, your task is to select a set of those roads to decorate, so that the parade groups will be able to travel to all the cities using these roads.

Input

The first line of the input contains the number of test cases T ($T \le 15$). For each test case:

- The first line consists of 2 integers N and M ($1 \le N \le 1000$, $1 \le M \le 30000$).
- Each of the next M lines describe an one-way road consisting of 3 integers u, v, c, which means that there is an one-way road from city u to city v and the cost to decorate this road is c ($c \le 1000000$).

Output

For each test case, print the total minimal cost in one line. Print -1 if there is no solution.

Sample

Sample input	Output for sample input
2 4 4 1 2 2 2 3 8 1 3 5 3 4 1 2 1 2 1 10	8 -1

Problem H – Array operations

A is a N-elements array, 1-based indexed. All elements of A have already been initialized. There are Q operations to apply on A. These operations are one of the two types:

- "SET i v" - set the value of A[i] to v ($|v| \le 10^9$)
- "GET l r c" - count how many elements A[i] where $l \le i \le r$ and $A[i] \le c$ ($|c| \le 10^9$)

Input

The first line of input contains the number of test cases - T ($T \le 10$). Each test case has the format:

- The first line consists of 2 integers N - the number of elements ($1 \le N \le 50000$) and Q - the number of operations ($1 \le Q \le 50000$).
- The second line consists of N integers which are the initial values of A.
- Then Q operations follow.

Output

Each test case should start with "Case #x" where x is the index of test case (numbered from 1). For each GET operation in the input, print the result in one line.

Sample

Sample input	Output for sample input
2 4 5 1 2 3 4 GET 1 4 4 GET 1 4 3 SET 1 5 GET 1 4 4 GET 1 4 3 5 5 1 2 3 4 5 GET 1 1 3 GET 1 2 3 GET 1 3 3 GET 1 4 3 GET 1 5 3	Case #1: 4 3 3 2 Case #2: 1 2 3 3 3

Problem I – Giant propeller

A giant propeller consists of N blades has just been built. Each blade is in one of the seven colors: red (R), orange (O), yellow (Y), green (G), blue (B), indigo (I) and violet (V). However, these blades are arranged randomly and do not follow any particular orders.

These blades should be re-arranged so that the spinning propeller will create a spectacular color effect. Two artists have been invited to design the propeller.

The first artist concludes: "If the blades are arranged according to rainbow-order color (ROYGBIV) in close-wise order, the effect would be great."

The second artist immediately says: "That would be boring! I want something more unnatural, just group them (the blades) by color. The order of 7 groups is not important."

A robot will be used to re-arrange these blades. Every second, the robot can swap any 2 consecutive blades. Given the chosen design and the current order of blades, determine the minimum time needed to re-arrange all the blades.

Input

The first line contains T - the number of test cases ($T \le 40$). Each test case will be printed in 2 lines. The first line consists of 2 integers N ($1 \le N \le 100000$) and the chosen design (1 or 2). The second line consists of the blades color in clockwise order.

Output

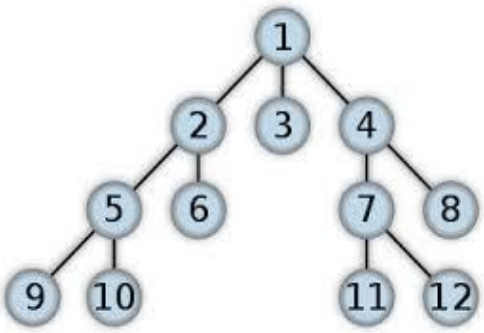
For each test case, print the minimum time to re-arrange all the blades.

Sample

Sample input	Output for sample input
3 7 1 VOYGIBR 8 2 ROYGBIVO 3 1 IOI	2 1 0

Problem J – Rooted tree

A tree is an undirected graph in which any two vertices are connected by *exactly one* path.



A tree is called a rooted tree if one vertex has been designated as the root, in which case the edges have a natural orientation, *towards* or *away from* the root.

In a rooted tree, the parent of a vertex is the vertex connected to it on the path to the root; every vertex except the root has a unique parent. A child of a vertex *v* is a vertex of which *v* is the parent.

A vertex is called leaf if it has no children.

The depth of a tree is the longest distance from the root to a leaf plus 1. In the example, the depth of the tree is 4.

We defined C[v] is the number of children of vertex v. In the example, C = {3, 2, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0}.

Given an array C, in general cases, we won't be able to reconstruct the structure of the tree but we are able to determine the possible range value of the depth of the tree - D. Your task is to determine this range.

Input

The first line of the input contains T - the number of test cases (T ≤ 100). Each test case is written in 2 lines.

- The first line contains N - the number of vertices (N ≤ 100).
- The second line contains N non-negative integers C[i]. The sum of C[i] is guaranteed to be equal to N-1.

Output

For each test case, print the result in one line. If you can determine exactly the value of D, print the value; otherwise, print the range in the format [min, max] (there is one space after the comma).

Sample

Sample input	Output for sample input
1 12 3 2 0 2 2 0 2 0 0 0 0 0	[4, 6]

Problem K – Buggy code

Robocon is a famous contest in robotics and automation which got a lot of attention from Vietnamese students. Team X from Bon Ba University is one of the strongest teams this year. Beside designing and manufacturing electrical circuits, they also prepared many different codes for different strategies.

In the final match, one of their robot starts at position 0 in the axis and need to travel to position N along the axis. Unfortunately, one of the team members mistakenly loaded a buggy code into the robot. Just when the referee starts the match, they immediately realize their mistake. Instead of an optimal travel from 0 to N, the robot travels in an unpredictable way. Every time the code asks the robot to move, the probability the robot moves one unit to the right (x to x+1) is p, the probability the robot moves one unit to the left (x to x-1) is (1 - p).

Now the team can do nothing but hope for their luck. Given N and p, your task is to calculate the expected number of moves for the robot to reach its goal.

Input

The first line contains the number of test cases T (T <= 100), followed by T test cases. Each test cases consists of an integer N ($0 < |N| \leq 1000$) and a real number p ($0 < p < 1$).

Output

For each test case, print the expected number of moves E with exactly 2 digits after decimal. Since the duration of the match is not that long, if E is equal or larger than one million, print “inf".

Sample

Sample input	Output for sample input
1 1 0.5	1.11