The 14th Zhejiang Provincial Collegiate Programming Contest

Sponsored by

图森 tu Simple

Contest Session

April 22, 2017

Problem List

This problem set should contain 13 (thirteen) problems on 20 (twenty) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.

https://sua.ac/

Hosted by

Sponsored by

Problem Set Prepared by

It's against the rules to open non-contest websites during the contest.
If you're interested (which is our pleasure), please scan
the QR code only after the contest.

Problem A. Cooking Competition

"Miss Kobayashi's Dragon Maid" is a Japanese manga series written and illustrated by Coolkyoushinja. An anime television series produced by Kyoto Animation aired in Japan between January and April 2017.

In episode 8, two main characters, Kobayashi and Tohru, challenged each other to a cook-off to decide who would make a lunchbox for Kanna's field trip. In order to decide who is the winner, they asked n people to taste their food, and changed their scores according to the feedback given by those people.

There are only four types of feedback. The types of feedback and the changes of score are given in the following table.

Type	Feedback	Score Change (Kobayashi)	Score Change (Tohru)
1	Kobayashi cooks better	$+1$	0
2	Tohru cooks better	0	$+1$
3	Both of them are good at cooking	$+1$	$+1$
4	Both of them are bad at cooking	-1	-1

Given the types of the feedback of these n people, can you find out the winner of the cooking competition (given that the initial score of Kobayashi and Tohru are both 0)?

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 100$), indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 20$), its meaning is shown above.

The next line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 4$), indicating the types of the feedback given by these n people.

Output

For each test case output one line. If Kobayashi gets a higher score, output "Kobayashi" (without quotes). If Tohru gets a higher score, output "Tohru" (without quotes). If Kobayashi's score is equal to that of Tohru's, output "Draw" (without quotes).

Example

standard input	standard output
3 3 1 2 1 2 2 4 2 3 4	Kobayashi Tohru Draw

Note

For the first test case, Kobayashi gets $1 + 0 + 1 = 2$ points, while Tohru gets $0 + 1 + 0 = 1$ point. So the winner is Kobayashi.

For the second test case, Kobayashi gets $0 - 1 = -1$ point, while Tohru gets $1 - 1 = 0$ point. So the winner is Tohru.

For the third test case, Kobayashi gets $1 - 1 = 0$ point, while Tohru gets $1 - 1 = 0$ point. So it's a draw.

Problem B. Problem Preparation

It's time to prepare the problems for the 14-th Zhejiang Provincial Collegiate Programming Contest! Almost all members of SUA programming contest problem setter team brainstorm and code day and night to catch the deadline, and empty bottles of *Marjar Cola* litter the floor almost everywhere!

To make matters worse, one of the team member fell ill just before the deadline. So you, a brilliant student, are found by the team leader Dai to help the team check the problems' arrangement.

Now you are given the difficulty score of all problems. Dai introduces you the rules of the arrangement:

1. The number of problems should lie between 10 and 13 (both inclusive).
2. The difficulty scores of the easiest problems (that is to say, the problems with the smallest difficulty scores) should be equal to 1.
3. At least two problems should have their difficulty scores equal to 1.
4. After sorting the problems by their difficulty scores in ascending order, the absolute value of the difference of the difficulty scores between two neighboring problems should be no larger than 2. BUT, if one of the two neighboring problems is the hardest problem, there is no limitation about the difference of the difficulty scores between them. The hardest problem is the problem with the largest difficulty score. It's guaranteed that there is exactly one hardest problem.

The team members have given you lots of possible arrangements. Please check whether these arrangements obey the rules or not.

Input

There are multiple test cases. The first line of the input is an integer T ($1 \le T \le 10^4$), indicating the number of test cases. Then T test cases follow.

The first line of each test case contains one integer n ($1 \le n \le 100$), indicating the number of problems.

The next line contains n integers $s_1, s_2, \ldots, s_n$ ($-1000 \le s_i \le 1000$), indicating the difficulty score of each problem.

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use scanf/printf instead of cin/cout in C++.

Output

For each test case, output "Yes" (without quotes) if the arrangement follows the rules, otherwise output "No" (without quotes).

Example

standard input	standard output
8 9 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 11 999 1 1 2 3 4 5 6 7 8 9 11 999 1 3 5 7 9 11 13 17 19 21 10 15 1 13 17 1 7 9 5 3 11 13 1 1 1 1 1 1 1 1 1 1 1 1 2 10 2 3 4 5 6 7 8 9 10 11 10 15 1 13 3 6 5 4 7 1 14	No No Yes No Yes Yes No No

Note

The first arrangement has 9 problems only, which violates the first rule.

Only one problem in the second and the fourth arrangement has a difficulty score of 1, which violates the third rule.

The easiest problem in the seventh arrangement is a problem with a difficulty score of 2, which violates the second rule.

After sorting the problems of the eighth arrangement by their difficulty scores in ascending order, we can get the sequence $\{1, 1, 3, 4, 5, 6, 7, 13, 14, 15\}$. We can easily discover that $|13-7| = 6 > 2$. As the problem with a difficulty score of 13 is not the hardest problem (the hardest problem in this arrangement is the problem with a difficulty score of 15), it violates the fourth rule.

Problem C. What Kind of Friends Are You?

Japari Park is a large zoo home to extant species, endangered species, extinct species, cryptids and some legendary creatures. Due to a mysterious substance known as *Sandstar*, all the animals have become anthropomorphized into girls known as *Friends*.

Kaban is a young girl who finds herself in Japari Park with no memory of who she was or where she came from. Shy yet resourceful, she travels through Japari Park along with Serval to find out her identity while encountering more *Friends* along the way, and eventually discovers that she is a human.

However, Kaban soon finds that it's also important to identify other *Friends*. Her friend, Serval, enlightens Kaban that she can use some questions whose expected answers are either "yes" or "no" to identitfy a kind of *Friend*.

To be more specific, there are n *Friends* need to be identified. Kaban will ask each of them q same questions and collect their answers. For each question, she also gets a full list of animals' names that will give a "yes" answer to that question (and those animals who are not in the list will give a "no" answer to that question), so it's possible to determine the name of a *Friend* by combining the answers and the lists together.

But the work is too heavy for Kaban. Can you help her to finish it?

Input

There are multiple test cases. The first line of the input is an integer T ($1 \le T \le 100$), indicating the number of test cases. Then T test cases follow.

The first line of each test case contains two integers n ($1 \le n \le 100$) and q ($1 \le q \le 21$), indicating the number of *Friends* need to be identified and the number of questions.

The next line contains an integer c ($1 \le c \le 200$) followed by c strings $p_1, p_2, \dots, p_c$ ($1 \le |p_i| \le 20$), indicating all known names of *Friends*.

For the next q lines, the i-th line contains an integer m_i ($0 \le m_i \le c$) followed by m_i strings $s_{i,1}, s_{i,2}, \dots, s_{i,m_i}$ ($1 \le |s_{i,j}| \le 20$), indicating the number of *Friends* and their names, who will give a "yes" answer to the i-th question. It's guaranteed that all the names appear in the known names of *Friends*.

For the following n lines, the i-th line contains q integers $a_{i,1}, a_{i,2}, \dots, a_{i,q}$ ($0 \le a_{i,j} \le 1$), indicating the answer (0 means "no", and 1 means "yes") to the j-th question given by the i-th *Friend* need to be identified.

It's guaranteed that all the names in the input consist of only uppercase and lowercase English letters.

Output

For each test case output n lines. If Kaban can determine the name of the i-th *Friend* need to be identified, print the name on the i-th line. Otherwise, print "Let's go to the library!!" (without quotes) on the i-th line instead.

Example

standard input	standard output
2 3 4 5 Serval Raccoon Fennec Alpaca Moose 4 Serval Raccoon Alpaca Moose 1 Serval 1 Fennec 1 Serval 1 1 0 1 0 0 0 0 1 0 0 0 5 5 11 A B C D E F G H I J K 3 A B K 4 A B D E 5 A B K D E 10 A B K D E F G H I J 4 B D E K 0 0 1 1 1 1 0 1 0 1 1 1 1 1 1 0 0 1 0 1 1 0 1 1 1	Serval Let's go to the library!! Let's go to the library!! Let's go to the library!! Let's go to the library!! B Let's go to the library!! K

Note

The explanation for the first sample test case is given as follows:

As Serval is the only known animal who gives a "yes" answer to the 1-st, 2-nd and 4-th question, and gives a "no" answer to the 3-rd question, we output "Serval" (without quotes) on the first line.

As no animal is known to give a "no" answer to all the questions, we output "Let's go to the library!!" (without quotes) on the second line.

Both Alpaca and Moose give a "yes" answer to the 1-st question, and a "no" answer to the 2-nd, 3-rd and 4-th question. So we can't determine the name of the third *Friend* need to be identified, and output "Let's go to the library!!" (without quotes) on the third line.

Problem D. Let's Chat

ACM (ACMers' Chatting Messenger) is a famous instant messaging software developed by Marjar Technology Company. To attract more users, Edward, the boss of Marjar Company, has recently added a new feature to the software. The new feature can be described as follows:

If two users, A and B, have been sending messages to **each other** on the last m **consecutive** days, the "friendship point" between them will be increased by 1 point.

More formally, if user A sent messages to user B on each day between the $(i-m+1)$-th day and the i-th day (both inclusive), and user B also sent messages to user A on each day between the $(i-m+1)$-th day and the i-th day (also both inclusive), the "friendship point" between A and B will be increased by 1 at the end of the i-th day.

Given the chatting logs of two users A and B during n consecutive days, what's the number of the friendship points between them at the end of the n-th day (given that the initial friendship point between them is 0)?

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 10$), indicating the number of test cases. For each test case:

The first line contains four integers n ($1 \le n \le 10^9$), m ($1 \le m \le n$), x and y ($1 \le x, y \le 100$). The meanings of n and m are described above, while x indicates the number of chatting logs about the messages sent by A to B, and y indicates the number of chatting logs about the messages sent by B to A.

For the following x lines, the i-th line contains two integers $l_{a,i}$ and $r_{a,i}$ ($1 \le l_{a,i} \le r_{a,i} \le n$), indicating that A sent messages to B on each day between the $l_{a,i}$-th day and the $r_{a,i}$-th day (both inclusive).

For the following y lines, the i-th line contains two integers $l_{b,i}$ and $r_{b,i}$ ($1 \le l_{b,i} \le r_{b,i} \le n$), indicating that B sent messages to A on each day between the $l_{b,i}$-th day and the $r_{b,i}$-th day (both inclusive).

It is guaranteed that for all $1 \le i < x$, $r_{a,i} + 1 < l_{a,i+1}$ and for all $1 \le i < y$, $r_{b,i} + 1 < l_{b,i+1}$.

Output

For each test case, output one line containing one integer, indicating the number of friendship points between A and B at the end of the n-th day.

Example

standard input	standard output
2 10 3 3 2 1 3 5 8 10 10 1 8 10 10 5 3 1 1 1 2 4 5	3 0

Note

For the first test case, user A and user B send messages to each other on the 1-st, 2-nd, 3-rd, 5-th, 6-th, 7-th, 8-th and 10-th day. As $m = 3$, the friendship points between them will be increased by 1 at the end of the 3-rd, 7-th and 8-th day. So the answer is 3.

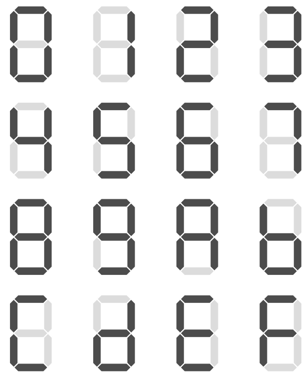
Problem E. Seven Segment Display

A seven segment display, or seven segment indicator, is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex dot matrix displays. Seven segment displays are widely used in digital clocks, electronic meters, basic calculators, and other electronic devices that display numerical information.

Edward, a student in Marjar University, is studying the course "Logic and Computer Design Fundamentals" this semester. He bought an eight-digit seven segment display component to make a hexadecimal counter for his course project.

In order to display a hexadecimal number, the seven segment display component needs to consume some electrical energy. The total energy cost for display a hexadecimal number on the component is the sum of the energy cost for displaying each digit of the number. Edward found the following table on the Internet, which describes the energy cost for display each kind of digit.

Digit	Energy Cost (units/s)	Digit	Energy Cost (units/s)
0	6	8	7
1	2	9	6
2	5	A	6
3	5	B	5
4	4	C	4
5	5	D	5
6	6	E	5
7	3	F	4



For example, in order to display the hexadecimal number "5A8BEF67" on the component for one second, $5 + 6 + 7 + 5 + 5 + 4 + 6 + 3 = 41$ units of energy will be consumed.

Edward's hexadecimal counter works as follows:

- The counter will only work for n seconds. After n seconds the counter will stop displaying.
- At the beginning of the 1-st second, the counter will begin to display a previously configured eight-digit hexadecimal number m.
- At the end of the i-th second ($1 \le i < n$), the number displayed will be increased by 1. If the number displayed will be larger than the hexadecimal number "FFFFFFFF" after increasing, the counter will set the number to 0 and continue displaying.

Given n and m, Edward is interested in the total units of energy consumed by the seven segment display component. Can you help him by working out this problem?

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 10^5$), indicating the number of test cases. For each test case:

The first and only line contains an integer n ($1 \le n \le 10^9$) and a capitalized eight-digit hexadecimal number m ($00000000 \le m \le \text{FFFFFFFF}$), their meanings are described above.

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use scanf/printf instead of cin/cout in C++.

Output

For each test case output one line, indicating the total units of energy consumed by the eight-digit seven segment display component.

Example

standard input	standard output
3 5 89ABCDEF 3 FFFFFFFF 7 00000000	208 124 327

Note

For the first test case, the counter will display 5 hexadecimal numbers (89ABCDEF, 89ABCDF0, 89ABCDF1, 89ABCDF2, 89ABCDF3) in 5 seconds. The total units of energy cost is

$$\begin{array}{ll}(7+6+6+5+4+5+5+4) & + \\ (7+6+6+5+4+5+4+6) & + \\ (7+6+6+5+4+5+4+2) & + \\ (7+6+6+5+4+5+4+5) & + \\ (7+6+6+5+4+5+4+5) & = \quad 208\end{array}$$

For the second test case, the counter will display 3 hexadecimal numbers (FFFFFFFF, 00000000, 00000001) in 3 seconds. The total units of energy cost is

$$\begin{array}{ll}(4+4+4+4+4+4+4+4) & + \\ (6+6+6+6+6+6+6+6) & + \\ (6+6+6+6+6+6+6+2) & = \quad 124\end{array}$$

Problem F. Heap Partition

A sequence $S = \{s_1, s_2, \ldots, s_n\}$ is called *heapable* if there exists a binary tree T with n nodes such that every node is labelled with exactly one element from the sequence S, and for every non-root node s_i and its parent s_j, $s_j \leq s_i$ and $j < i$ hold. Each element in sequence S can be used to label a node in tree T only once.

Chiaki has a sequence $a_1, a_2, \ldots, a_n$, she would like to decompose it into a minimum number of heapable subsequences.

Note that a subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contain an integer n ($1 \leq n \leq 10^5$) – the length of the sequence.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$).

It is guaranteed that the sum of all n does not exceed $2 \cdot 10^6$.

Output

For each test case, output an integer m denoting the minimum number of heapable subsequences in the first line. For the next m lines, first output an integer C_i, indicating the length of the subsequence. Then output C_i integers $P_{i1}, P_{i2}, \ldots, P_{iC_i}$ in increasing order on the same line, where P_{ij} means the index of the j-th element of the i-th subsequence in the original sequence.

Example

standard input	standard output
4 4 1 2 3 4 4 2 4 3 1 4 1 1 1 1 5 3 2 1 4 1	1 4 1 2 3 4 2 3 1 2 3 1 4 1 4 1 2 3 4 3 2 1 4 1 2 2 3 5

Problem G. Yet Another Game of Stones

Alice and Bob are playing yet another game of stones. The rules of this game are as follow:

- The game starts with n piles of stones indexed from 1 to n. The i-th pile contains a_i stones and a special constraint indicated as b_i.
- The players make their moves alternatively. **The allowable moves for the two players are different**.
- An allowable move of Bob is considered as removal of some positive number of stones from a pile.
- An allowable move of Alice is also considered as removal of some positive number of stones from a pile, but is limited by the constraint b_i of that pile.
  - If $b_i = 0$, there are no constraints.
  - If $b_i = 1$, Alice can only remove some odd number of stones from that pile.
  - If $b_i = 2$, Alice can only remove some even number of stones from that pile.

  Please note that there are no constraints on Bob.
- The player who is unable to make an allowable move loses.

Alice is always the first to make a move. Do you know who will win the game if they both play optimally?

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$), indicating the number of piles.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), indicating the number of stones in each pile.

The third line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($0 \le b_i \le 2$), indicating the special constraint of each pile.

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use scanf/printf instead of cin/cout in C++.

Output

For each test case, output "Alice" (without the quotes) if Alice will win the game. Otherwise, output "Bob" (without the quotes).

Example

standard input	standard output
3 2 4 1 1 0 1 3 2 1 1 2	Alice Bob Bob

Note

For the first test case, Alice can remove 3 stones from the first pile, and then she will win the game.

For the second test case, as Alice can only remove some even number of stones, she is unable to remove all the stones in the first move. So Bob can remove all the remaining stones in his move and win the game.

For the third test case, Alice is unable to remove any number of stones at the beginning of the game, so Bob wins.

Problem H. Binary Tree Restoring

Given two depth-first-search (DFS) sequences of a binary tree, can you find a binary tree which satisfies both of the DFS sequences?

Recall that a binary tree is a tree in which each vertex has at most two children, and the depth-first search is a tree traversing method which starts at the root and explores as far as possible along each branch before backtracking.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$), indicating the number of vertices in the binary tree.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$, $\forall 1 \le i < j \le n, a_i \ne a_j$), indicating the first DFS sequence of the binary tree.

The third line of each test case contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le n$, $\forall 1 \le i < j \le n, b_i \ne b_j$), indicating the second DFS sequence of the binary tree.

It is guaranteed that the sum of n over all test cases does not exceed 10^6, and there always exists at least one possible binary tree.

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use scanf/printf instead of cin/cout in C++.

Output

For each test case, output one line which contains n integers seperated by one space. The i-th integer indicates the father of the i-th vertex in the binary tree which satisfies both of the DFS sequence. If the i-th vertex is the root of the binary tree, output 0 as its father. If there are multiple valid answers, you can output any of them.

Please, DO NOT print extra spaces at the end of each line, or your program may get a "wrong answer" verdict as this problem is special judged.

Example

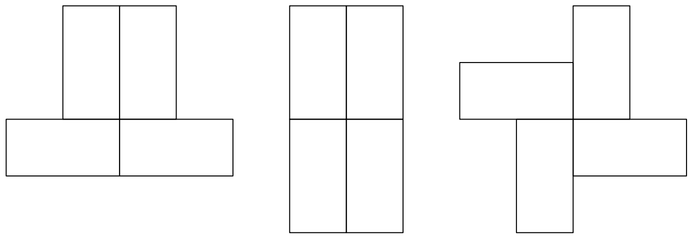
standard input	standard output
2 6 3 4 2 5 1 6 3 4 5 2 1 6 3 1 2 3 1 2 3	3 4 0 3 4 1 0 1 2

Problem I. Domino Tiling

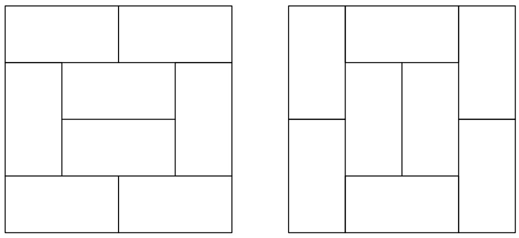
Chiaki has an $n \times m$ rectangular chessboard. She would like to tile this board with dominoes, where a domino is a 2×1 rectangle, such that:

- all the squares of the board are covered but no dominoes overlap or lie partially off the board.
- there must be no points where corners of four different dominoes meet.

The figure below shows some forbidden configurations:



The figure below shows two valid tilings of 4×4 chessboard:



You also need to number the dominoes of chessboard so that no two dominoes have the same number. You can use the number from 1 to $n \times m$.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \leq n, m \leq 100$) – the size of the rectangular chessboard.

It is guaranteed that the sum of $n \times m$ over all test cases does not exceed 2×10^6.

Output

For each test case, output a valid chessboard described above. A valid chessboard consists of n lines and each line contains m integers. Each integer in the output should represent the *id* of a domino. The grids sharing the same *id* belong to the same domino.

If there is no solution, output "Impossible!" (without the quotes) instead.

Example

standard input	standard output
3 1 1 4 3 4 4	Impossible! 1 1 2 3 4 2 3 4 5 6 6 5 1 1 2 2 3 4 4 5 3 6 6 5 7 7 8 8

Problem J. Card Game

Alice and Bob are playing games again. This game has nothing to do with stones. It is actually a card game.

There are n cards on the table, each with two integers (one is red, and the other one is blue) written on it. At the beginning of each round, two integers, L and R, will be given. Alice will pick an integer x such that $L \le x \le R$ and tell her choice to Bob. After knowing the integer x, Bob will then choose a card from the table. The score of this round will be equal to $rx + b$, where r is the red integer on the chosen card and b is the blue integer on the chosen card.

Both Alice and Bob are free to check the cards on the table and the integers written on them before they make their decisions.

To make the game more interesting, some changes can be made before a certain round. There are two possible changes:

1. Add another card with a red integer and a blue integer written on it onto the table.
2. Remove a card from the table.

Alice wants to maximize the score of each round, while Bob wants to minimize it. If both of them play the game optimally, can you find out the final score for each round?

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integer n ($1 \le n \le 5 \times 10^4$) and q ($1 \le q \le 5 \times 10^4$), indicating the number of cards on the table at the beginning, and the number of operations.

For the following n lines, the i-th line contains two integers r_i and b_i ($-10^9 \le r_i, b_i \le 10^9$), indicating that initially there is a card with a red integer r_i and a blue integer b_i written on it on the table.

For the following q lines, each line contains three integers op ($0 \le op \le 2$), a and b ($-10^9 \le a, b \le 10^9$).

- If op equals 0, you are asked to calculate the final score of a round, where $L = a$ and $R = b$ is given. It's guaranteed that $a \le b$ on this occasion.
- If op equals 1, a card with a red integer a and a blue integer b written on it will be put onto the table.
- If op equals 2, a card with a red integer a and a blue integer b written on it will be removed from the table. It's guaranteed that this card exists on the table. If there are multiple cards on the table which satisfy the condition, only one of them will be removed.

It's guaranteed that neither the sum of n nor the sum of q over all test cases will exceed 2×10^5.

We kindly remind you that this problem contains large I/O file, so it's recommended to use a faster I/O method. For example, you can use scanf/printf instead of cin/cout in C++.

Output

For each operation 0 output one line, indicating the final score of that round.

Example

standard input	standard output
2 2 7 -1 2 2 3 0 -1 1 2 -1 2 0 -1 1 2 2 3 1 1 2 1 -2 -1 0 -1 1 2 3 1 1 1 1 0 1 3 2 1 1 0 1 3	2 5 1 4 4

Problem K. Final Defense Line

There is a circle in the plane. Both the coordinates of the center and the radius are unknown.

Chiaki found three distinct points A, B and C in the plane. And she also knows the shortest distance from each point to the circumference.

Chiaki would like to find the smallest circle according to above information.

Note that in general, a circle with infinite radius is a line. But in this problem, line is not considered as a circle.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 2 \times 10^5$), indicating the number of test cases. For each test case:

The first line contains three integers x_a, y_a and d_a ($-100 \le x_a \le 100, y_a = 0, 1 \le d_a \le 100$) denoting the coordinates of A and the shortest distance to the circumference.

The second line contains three integers x_b, y_b and d_b ($-100 \le x_b \le 100, y_b = 0, 1 \le d_b \le 100$) denoting the coordinates of B and the shortest distance to the circumference.

The third line contains three integers x_c, y_c and d_c ($-100 \le x_c, y_c, d_c \le 100, d_c \ne 0$) denoting the coordinates of C and the shortest distance to the circumference.

If the distance is equal to 0, the point is on the circumference. If distance is greater than 0, the point is inside the circle. If distance is less than 0, the point is outside the circle and the shortest distance is the absolute value.

It is guaranteed that the minimum possible radius of the circle is at most 10^4.

Output

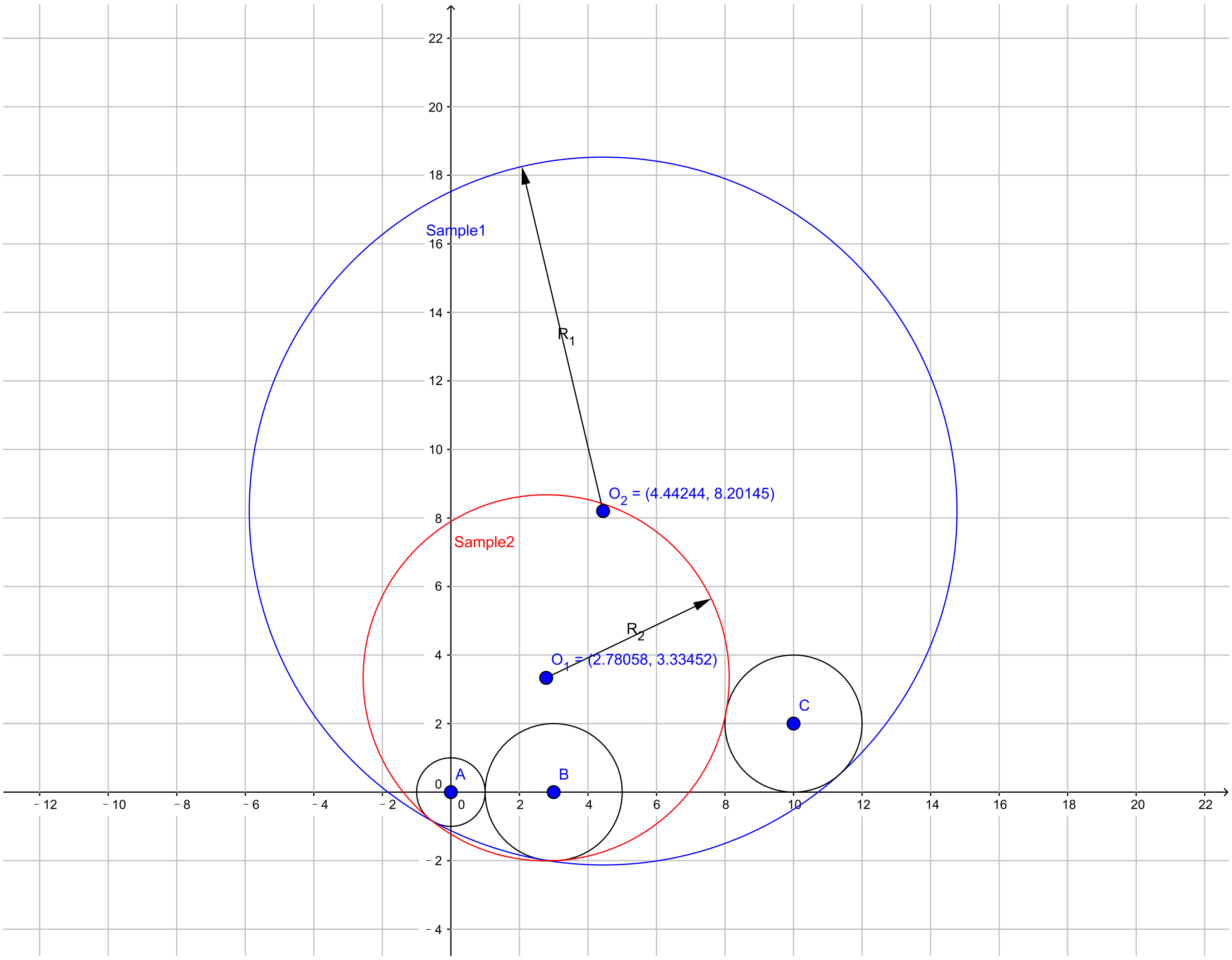
For each test case, if there are infinite possible circles, output -1 in a single line. If there is no such circle, output 0 in a single line. Otherwise, output an integer m and a real number r in a single line separated by one space denoting the number of possible circles and the radius of the smallest circle. You answer will be accepted if the relative error of your answer is no more than 10^{-6}.

Example

standard input	standard output
2 0 0 1 3 0 2 10 2 2 0 0 1 3 0 2 10 2 -2	2 10.327329213474 2 5.341730785446

Note

The image below shows the sample.


Sample1
Sample2
R1
R2
O2 = (4.44244, 8.20145)
O1 = (2.78058, 3.33452)
A
B
C

Problem L. Chiaki Sequence

Chiaki is interested in an infinite sequence $a_1, a_2, a_3, \ldots$, which defined as follows:

$$a_n = \begin{cases} n & n \le 2 \\ 2 \cdot a_{n-1} & n \text{ is odd} \\ a_{n-1} + r_{n-1} & n \text{ is even} \end{cases}$$

where r_n is the smallest positive integer not in the set $S_n = \{a_j - a_i | 1 \le i < j \le n\}$.

Chiaki would like to know the sum of the first n terms of the sequence, i.e. $\sum\limits_{i=1}^{n} a_i$. As this number may be very large, Chiaki is only interested in its remainder modulo $(10^9 + 7)$.

Input

There are multiple test cases. The first line of input contains an integer T ($1 \le T \le 1000$), indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n < 10^{100}$) without leading zeros.

Output

For each test case, output an integer denoting the answer.

Example

standard input	standard output
11 1 2 3 4 5 6 7 8 9 10 1000000000	1 3 7 15 31 52 94 145 247 359 834069170

Problem M. Sequence to Sequence

Chiaki has a sequence $s_1, s_2, \ldots, s_n$. She would like to change it to another sequence $t_1, t_2, \ldots, t_n$ using the following operations:

- choose two indices l and r ($l \le r$), and add 1 to every nonzero element between the indices l and r (both inclusive).
- choose two indices l and r ($l \le r$), and subtract 1 from every nonzero element between the indices l and r (both inclusive).

Chiaki would like to know the minimum number of operations needed.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$) – the length of the sequence.

The second line contains n integers $s_1, s_2, \ldots, s_n$ ($0 \le s_i \le 10^9$).

The third line contains n integers $t_1, t_2, \ldots, t_n$ ($0 \le t_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed 10^6.

Output

For each test case, output an integer denoting the minimum number of operations. If it is impossible to change the sequence, output -1 instead.

Example

standard input	standard output
2 5 1 1 1 1 1 2 0 2 0 2 7 3 1 2 3 2 1 4 2 0 0 0 0 0 2	3 3

Note

For the first test case: $\{1,1,1,1,1\} \xrightarrow{[2,2],\ -1} \{1,0,1,1,1\} \xrightarrow{[4,4],\ -1} \{1,0,1,0,1\} \xrightarrow{[1,5],\ +1} \{2,0,2,0,2\}$.