Problem A. [redacted]

Program:	`poetry.(c|cpp|java)`
Input:	`poetry.in`
Balloon Color:	`Green`

Let's check another challenge of the IBM ICPC Chill Zone, a poetry challenge. One says a poetry string that starts with an English letter and ends with an English letter, the second should say a poetry string that starts with the same letter that the previous string ended with.

Given the two poetry string sets representing the known strings for each player. Each player can use each of his strings only once. If during the player turn he can not say any string, he loses. Assuming both players play optimally well determine which player wins the game depending on the given two sets.

Input

The first line contains an integer T represent the number of the following test cases.
Each test case starts with an integer n the number of strings in the first player set. Each of the next n lines contains a string of the first player set. Then read an integer m, which will be succeeded by m lines describing the strings of the second player. No string in the input will start or finish with a white space, only lowercase letters. The length of each string in the input will not exceed 10,000 letters.

$1 \leq n \leq 9$
$1 \leq m \leq 9$
$1 \leq T \leq 10$

Output

For each test case, print one line saying which player should win if they are so clever to play it perfectly and assuming that each one knows the set of the other player.

Discarding quotes, print "Game_i:_player1"to denote the wining of the first player or "Game_i:_player2"to denote the win of the second player where 'i' represents the game number starting from 1. Replace the underscores with spaces.

Examples

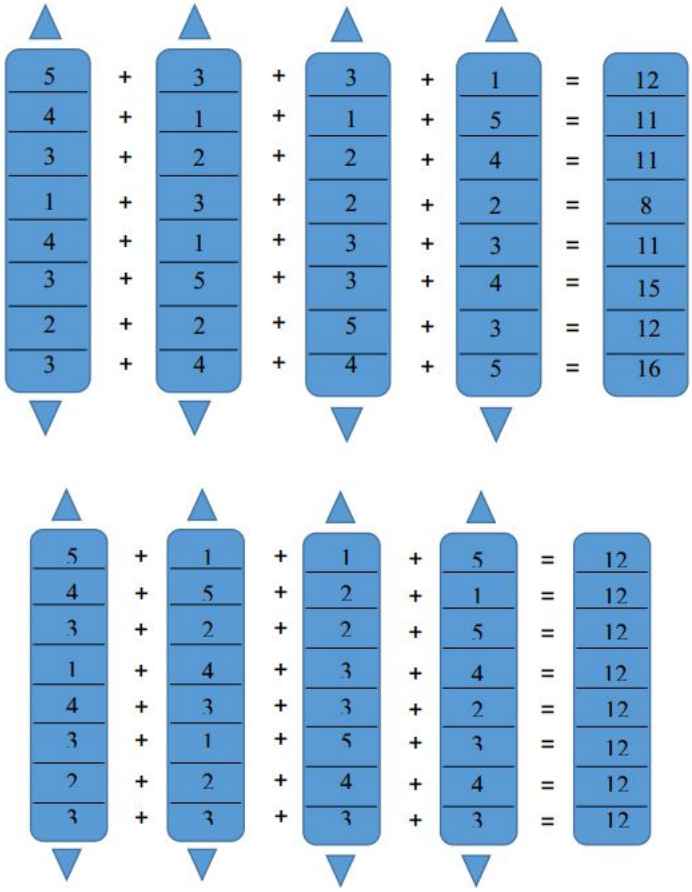
poetry.in	Standard Output
2 3 a poetry string a poetry string starting with a a poetry string ending with a 3 generated word a word ending with b poetry 2 either one or two random string 3 another test case one greatest poetry be the winner	Game 1: player2 Game 2: player1

Problem B. [redacted]

Program:	`room.(c|cpp|java)`
Input:	`room.in`
Balloon Color:	`Rose`

One of the most committed, loyal, and hard-working volunteers in the group ACPC organizers is Loubna Benguit. She showed real dedication and full commitment to the noble cause of the ICPC competition from her day 1. Loubna is from Morocco. She has just graduated from École nationale supérieure d'informatique et d'analyse des systèmes, last July. Last year was the final year of Loubna's undergraduate journey, she was supposed to deliver her final graduation project presentation by mid June, but as you know Loubna was part of the ACPC team who traveled to Russia to attend the World Finals event.

Loubna was super lucky to have Dr Sidi Ali with her in Russia who was the one directly responsible for grading her graduation project presentation, therefor she decided to present her graduation project there. Coach Fegla reserved a room in Onegin Hotel in Yekaterinburg, Russia for the purpose of the presentation. The next day when Coach Fegla and Loubna went to the room to prepare it for the presentation they found out that it was locked. The lock was designed by Dr Sidi Ali as a test for Loubna, and it can only be unlocked by a true programmer. The lock consisted of 4 sliders that can go either up or down like stated in the image below. Each slider contains the same number of integers. You have to slide the four sliders up or down to produce an equal sum for all the rows, like in the second image.



In the image above, you have to rotate the 2nd slider 4 times upward, the 3rd slider 1 time upward, and the 4th slider 1 time downward. This way you can make the sum of each row equal to 12. Note that you have made a total of 6 rotations.

Can you help Loubna and Fegla, to determine if it is possible to open the lock?

Input

The first line will be the number of test cases T. Each test case will start with an integer N, which represents the number of integers on each of the sliders. Then it will be followed by 4 lines each line containing N integers. The i_{th} line $1 \leq i \leq 4$ will have the N integers describing the i_{th} slider.

$1 \leq T \leq 100$
$1 \leq N \leq 1000$
$1 \leq$ *a single value within a slider* $\leq 4,000,000$

Output

For each test case print a single line containing:
Case_x:_y

x is the case number starting from 1.
y is "Yes"if the the lock can be opened, otherwise "No without the quotes.
Replace all underscores with spaces.

Examples

room.in	Standard Output
3 8 5 4 3 1 4 3 2 3 3 1 2 3 1 5 2 4 3 1 2 2 3 3 5 4 1 5 4 2 3 4 3 5 8 8 9 1 9 1 8 4 3 9 5 2 5 9 3 7 3 4 3 2 8 8 1 7 9 1 2 6 1 5 3 6 5 8 1 7 2 2 3 1 1 19 2 22 23 21 6 7 4 7 21 5 3 18 4 4 5 20 5 4 4 6 4 5 22 6	Case 1: Yes Case 2: No Case 3: Yes

Problem C. [redacted]

Program:	`giveaways.(c|cpp|java)`
Input:	`giveaways.in`
Balloon Color:	`Red`

During the preparation for the ICPC contest, the organizers prepare bags full of giveaways for the contestants. Each bag usually contains an MP3 Player, a Sim Card, a USB HUB, a USB Flash Drive, ... etc. A problem happened during the delivery of the components of the bags, so not every component was delivered completely to the organizers. For example the organizers ordered 10 items of 4 different types, and what was delivered was 7, 6, 8, 9 from each type respectively.

The organizers decided to form bags anyway, but they have to abide by 2 rules. All formed bags should have exactly the same items, and no bag should contain 2 items of the same type (either 1 or 0).

Knowing that each item has an amusement value (for sure an MP3 Player is much more amusing than a Sim Card), the organizers decided to get the max possible total amusement. The total amusement is the amusement value of a single bag multiplied by the number of bags. Note that not every contestant should receive a bag.

The amusement value of each item type is calculated using this equation:$(i \times i) \bmod C$
where i is an integer that represents the item type, and C is a value that will be given as an input.

Please help the ICPC organizers to determine what the maximum total amusement is.

Input

T is the number of test cases. For each test case there will be 3 integers M,N and C, where M is the number of items, N is the total number of types, and C is as described above then M integer representing the type of each item.

$1 \leq T \leq 100$
$1 \leq M \leq 10,000$
$1 \leq N \leq 10,000$
$1 \leq C \leq 10,000$
$1 \leq item_i \leq N$

Output

For each test case print a single line containing: Case_x:_y

x is the case number starting from 1. y is the required answer.
Replace the underscores with spaces.

Examples

giveaways.in	Standard Output
1 10 3 9 1 1 2 2 1 1 2 3 1 2	Case 1: 20

Problem D. [redacted]

Program:	`tshirts.(c|cpp|java)`
Input:	`tshirts.in`
Balloon Color:	`Pink`

It was the second day of IBM Chill Zone, and it was the time for distributing the prizes. Unfortunately due to unknown reasons, the organizing committee can only buy T-shirts to the contestants or give them D dollars in cash. The T-shirts factory only permitted them to order a single bulk of T-shirts of the same size where a single T-shirt price was determined using this equation:
$C \times S$
Where C is a given constant, and S is the size of the T-shirt.

There was only one limitation on the T-shirts they will give to the contestants, any contestant only accepts to receive a T-shirt of greater or equal size but not smaller size.

The organizing committee decided to minimize the money they should pay . Please help them in determining the amount they need to pay in order to give each contestant either a T-shirt or D dollars.

Input

T the number of test cases. For each test case there will be three integers N, D and C , then N integers representing the size of each of the T-shirt of each of the contestants.

$1 \leq T \leq 100$
$1 \leq N \leq 100,000$
$1 \leq D \leq 10,000$
$1 \leq C \leq 10,000$
$1 \leq t - shirtsize \leq 100,000$

Output

For each test case print a single line containing: Case_x:_y

x is the case number starting from 1. y is is the required answer.
Replace the underscores with spaces.

Examples

tshirts.in	Standard Output
1 5 100 1 35 70 75 90 110	Case 1: 425

Note

The optimal solution is to buy 3 tshirts of size 75 for the first 3 contestants, and give 100 dollars to the last two.

Problem E. [redacted]

Program:	`zone.(c|cpp|java)`
Input:	`zone.in`
Balloon Color:	`Orange`

A relaxing, fun way to unwind nightly with old and new friends at the ACM-ICPC World Finals is to stop by the IBM Chill Zone! A great way to participate in interactive games and interesting conversation with innovative IBMers and attendees from all over the world, the IBM Chill Zone is always a favorite for all. Many games can be found in the chill zone, including board game, human-size chess board, real-life angry birds, and stones game.

The stones game was invented by one of ICPC world finalists. It a 2-player game. It consists of a line of n stones and at each move a player should remove k consecutive stones, and if the player can not make any moves he loses, for example if we have $n = 6$, $k = 2$ (stones are represented by '*', and empty spaces by '.'):

STARTING STATE	* * * * * *
FIRST PLAYER	* . . * * *
SECOND PLAYER	* . . . . *

First player now has no valid moves, so he loses, but note that he did not play optimally here.

Given n, and k assume both players play optimally well, determine if the first player is losing or winning.

Input

The first line will be the number of test cases T. Each of the following T lines will contain 2 numbers n, k.

$1 \leq T \leq 100$
$1 \leq n \leq 50$
$1 \leq k \leq n$

Output

For each test case print a single line containing:
Case_x:_y

x is the case number starting from 1.
y is either 'Winning', or 'Losing' without the quotes (winning if the first is winning, losing otherwise)
Replace underscores with spaces.

Examples

zone.in	Standard Output
2 5 2 5 3	Case 1: Losing Case 2: Winning

Problem F. [redacted]

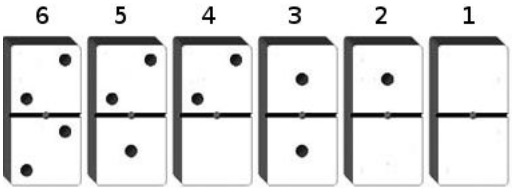
Program: `lock.(c|cpp|java)`
Input: `lock.in`
Balloon Color: `White`

It was a beautiful night at Yekaterinburg, the sky was crystal clear with no clouds, and the view of the moon and the stars was magnificent. Coach Fegla was enjoying this view at the Novotel hotel with Hanaa El-Jazzar (a member of the systems operation team of the ACPC). It was the first time for Hanaa to attend the ICPC world finals event, and she was so happy about it and brought coach Fegla a gift from her hometown Lebanon as a sign of gratitude. The gift was a round shaped plate with engraved images of different sightseeing in Lebanon.

While Hanaa was showing the present to the coach, two drunk fellows entered the room by mistake. They were so high they could not distinguish their room, from the coach's room. Coach Fegla thought what if the ones who entered the room were complete strangers, what should he do? He decided to install another lock to his door, but it was not like any other lock. To obtain the key of the lock a certain puzzle has to be solved, this way coach Fegla might be able to stop most of the intruders.

The puzzle was as follows, you will be given 2 numbers n and m. You should build up a domino set. Each piece of the set will contain 2 integer numbers in the range from 0 to n-1, and all the pieces are pairwise distinct and no two pieces of domino should look the same if rotated. As you know a domino piece is divided into 2 halves where the top half contains one integer and other half contains the other integer. You should lay down the pieces in a straight line where the top half of each piece should contain the greatest value of the 2 halves. Then you should sort the domino pieces in increasing order based on the top half, in case of equality sort them in increasing order based on the other half. The lock to the key will be the m_{th} piece in this sequence.

You will be given n, m can you get lock key? (check the image below for a domino set where n = 3)



Input

The first line will be the number of test cases T. Each of the following lines will contain 2 numbers n, m.

$1 \leq T \leq 100{,}000$
$1 \leq n \leq 1{,}000{,}000{,}000$
$1 \leq m \leq \frac{n \times (n+1)}{2}$

Output

For each test case print a single line containing:
Case_x:_a_b

x is the case number starting from 1.
a is the greatest value on the domino piece, and b is the other value.
Replace underscores with spaces.

Examples

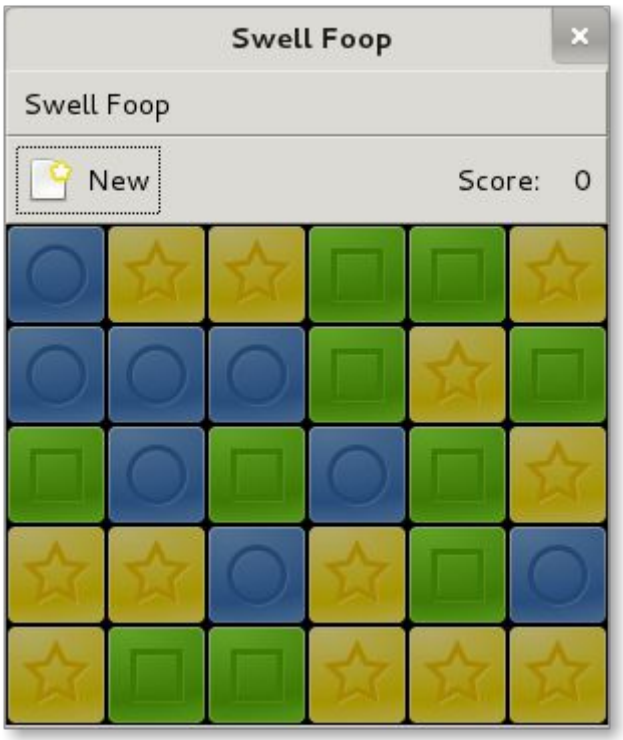
lock.in	Standard Output
2 2 1 3 5	Case 1: 0 0 Case 2: 2 1

Problem G. [redacted]

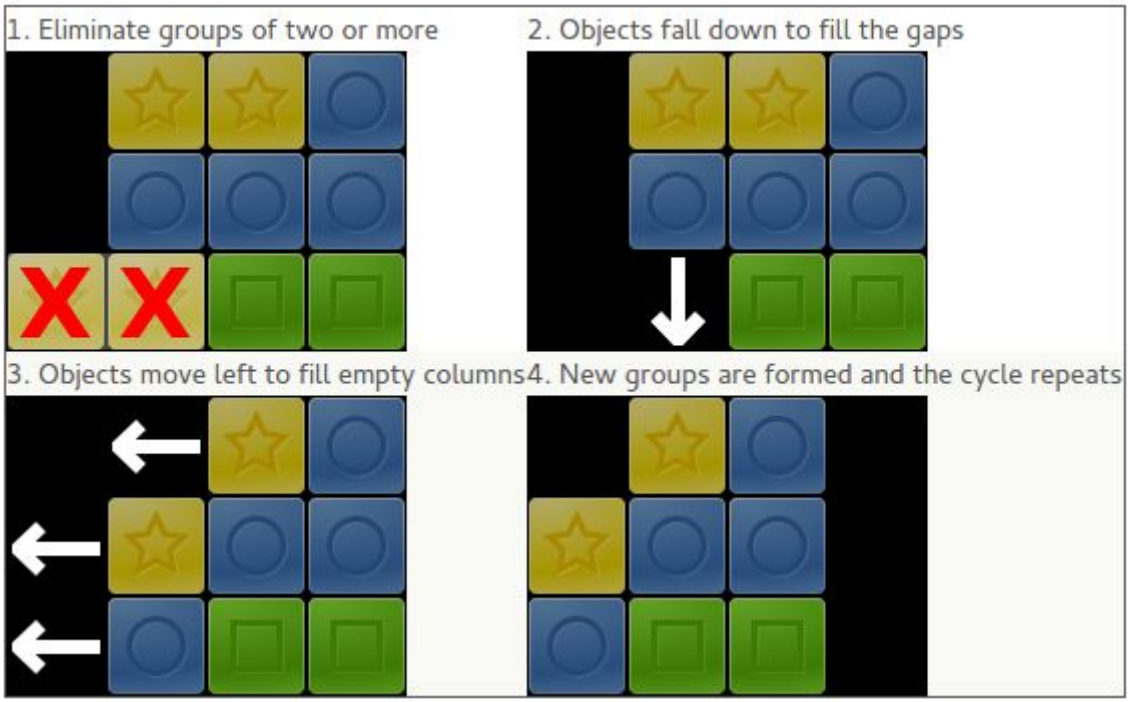
Program:	`swellfoop.(c|cpp|java)`
Input:	`swellfoop.in`
Balloon Color:	`Dark Purple`

During the 5 hours of the ICPC world finals contest, coach Fegla was playing the Swell Foop game on Ubuntu, an amazing and time consuming game, that kept him engaged for the whole time. He was trying to maximize his score, but it was so difficult as the game has so many possibilities. After the contest he started working on a program to solve the game, but he was quite tight on time as he should prepare for the trip back to Cairo. Can you please help him writing this program?

Swell Foop is a puzzle game. The goal is to remove as many objects as possible in as few moves as possible. Objects that are adjacent to each other get removed as a group. The remaining objects then collapse to fill in the gaps and new groups are formed. You cannot remove single objects.



The board starts as a full grid of objects. Depending on the size of board you select, there are either three or four types of object. If a group of objects are adjacent and all the same type, then they can be removed simply by clicking them with the mouse (or by pressing the space bar). When you move the mouse over a group you can remove, the objects will start moving. The number of objects in the group, and the points you will score for removing that group, is shown below. The more objects in the group, the more points you will score. Once the group has been removed the objects above them fall down to fill the space. If an entire column is cleared then the objects slide leftward to fill the gap.



Scoring is based on the number of objects you delete:

Number of Objects	Points Scored
2	0
3	1
4	4
5	9
6	16
7	25
8	36
9	49
⋮	⋮
n	(n - 2)2

If you clear the board there is a 1000 point bonus.

The task here is really simple, Given a full description of the initial grid, can you remove all the objects using the the rules stated above?

Input

The first line will be the number of test cases T. Each test case consists of 5 rows and 6 columns. The i_th row and j_th column should have the color of this cell expressed in terms of a single char where 'R' means red, 'B' means blue, 'Y' means yellow and 'G' means green, no other characters will be presented.

$1 \leq T \leq 100$

Output

For each test case print a single line containing:
Case_x:_y

x is the case number starting from 1.

y is the required answer, "Yes"if possible, "No"otherwise.
Replace underscores with spaces.

Examples

swellfoop.in	Standard Output
2 RRBBGG GGRRBB BBGGRR RRBBGG GGRRBY RRBBGG GGRRBB BBGGRR RRBBGG GGRRBB	Case 1: No Case 2: Yes

Problem J. [redacted]

Problem H

Program:	bye.(c|cpp|java)
Input:	bye.in
Balloon Color:	Cyan

It was the last day at Russia, after the World Finals has ended. Coach fegla was travelling back to Cairo on the same day but later. Coach Fegla was really tired, and he had to sleep before he headed to the Airport.

Due to an unknown reason his phone alarm was not working, and he needed something to wake him up. Fortunately his stopwatch was just working fine, and it worked in count down mode, and it supported only minutes. You will be given the current time, and the time he wished to wake up, help him determine the number of needed minutes to configure the stopwatch properly.

Input

The first line will be the number of test cases T. The following T lines each will contain 4 integers hc mc hw mw which represent the current time(hours, and minutes), and the wake up time(hours, minutes).

$1 \leq T \leq 100$
$0 \leq hc, hw \leq 23$
$0 \leq mc, mw \leq 59$

Note that both times will be on the same day, and the wake up time comes after the current time. Assume the day starts at 00:00 and ends at 23:59.

Output

For each test case print a single line containing: Case_x:_y

x is the case number starting from 1. y is the required answer.
Replace underscores with spaces.

Examples

bye.in	Standard Output
2 1 22 13 21 2 35 21 35	Case 1: 719 Case 2: 1140