Problem A. The Friends

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

John and Brus are the best friends. They spend almost all the time together playing games, watching TV, listening to the music, solving algorithmic problems etc. But recently something terrible has happened and now the best friends don't talk to each other.

John: We are still the best friends, aren't we?

Brus: Okay, I'll give you a chance!

The problem is about a girl. Each of two guys would like to date her, but unfortunately she can't date both of them at the same time. So, John and Brus decide to solve this problem in the following way. Each of them will generate some integer number using his random generator machine. John will generate some random integer between J_1 and J_2 inclusive. Let's denote that number with J. Similarly Brus will randomly generate some integer between B_1 and B_2 inclusive. We will denote it with B. Further, if J^B (J to the power of B) is greater than B^J (B to the power of J), then John will date the girl and if B^J is greater than J^B then Brus will be the lucky winner. In case of a tie they will repeat the procedure from the very beginning. You have to calculate the winning chances for John and Brus.

Input

The first line contains single integer T — the number of test cases ($1 \le T \le 47$). Each test case consists of a single line containing four integers J_1, J_2, B_1 and B_2 ($1 \le J_1 < J_2 \le 10^9$, $1 \le B_1 < B_2 \le 10^9$) in this very order. All the integers are separated by single spaces.

Output

For each test case print a single line containing probabilities P_J and P_B separated by a single space. Here P_J is John's chance to date the girl and P_B is the chance for Brus. P_J and P_B should be represented as irreducible fractions in `a/b` form, where $0 \le A$ and $b > 0$.

Example

standard input	standard output
2 1 3 2 4 4 7 44 77	1/3 2/3 1/1 0/1

Problem B. The Cities

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 Mebibytes

After dating the girl John and Brus finally got together again. Now they are enjoying the sweet memories about the good times when they were travelling together around the world.

John: Brus, have you ever been to Albania?

Brus: No, but I guess it should be a beautiful city...

There are N cities visited by John and Brus. John wrote down these cities. And now, while there are at least three cities in the list, Brus will randomly choose three of them and place the marks on the world map. Then John will calculate the area of the triangle formed by these cities and cross them out of the list. At the end Brus will sum up all the areas calculated by John. You have to find an expected value of this sum. Note that world map is flat and some triangles may be degenerate.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case starts with a line containing single integer N ($1 \le N \le 1000$). The next N lines contain coordinates of the cities visited by our guys — X_i and Y_i ($-1000 \le X_i, Y_i \le 1000$). All (X_i, Y_i) will be distinct. Each pair of coordinates is separated by a single space.

Output

For each test case print a single line containing the expected value of the sum calculated by Brus. Your answers must be rounded (using standard rounding rules) to seven digits after the decimal point.

Example

standard input	standard output
2 7 -3 5 -10 -8 6 10 8 6 8 3 -1 8 4 4 2 4 7 9 -1	58.8285714 0.0000000

Problem C. The Unfair Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

John and Brus are playing the unfair game on $H \times W$ board. John has a single checker which is initially located at $(R_J,\ C_J)$. And Brus has a single checker which is initially located at $(R_B,\ C_B)$. All coordinates are 1-based. The two players alternate turns.

When it is John's turn, he chooses one of four directions (up, down, left or right) and moves his checker one cell in the chosen direction. When it is Brus's turn, he also chooses one of those four directions and moves his checker any positive number of cells in the chosen direction. A player wins the game when his move puts his checker in the cell occupied by his opponent's checker.

John: Brus, do you really think it's fair?

Brus: Sure, just go on!

Both players use an optimal game strategy. If the player can win, he will follow the strategy that minimizes the number of moves in the game. If the player can't win, he will follow the strategy that maximizes the number of moves in the game. You have to determine the game outcome.

Input

The first line contains single integer T $(1 \le T \le 47)$ — the number of test cases. Each test case starts with a line containing two integers H and W. The following line contains four integers R_J, C_J, R_B and C_B $(1 \le R_J, R_B \le H,\ 1 \le C_J, C_B \le W)$, in this very order. (R_J, C_J) and (R_B, C_B) will be different. All the integers in a single line are separated by single spaces. The next line contains the name of the player (either John or Brus) who will start the game.

Output

For each test case print a single line containing the name of the winner and the total number of moves in the game separated by a single space.

Example

standard input	standard output
2 4 7 1 1 2 7 John 2 2 1 2 2 1 Brus	Brus 6 John 2

Problem D. The Almost Lucky Numbers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Of course, this is the fourth problem, thus it should be about the lucky numbers.

John: Brus, Brus! I almost did it!

Brus: So, you are almost lucky!

Almost everyone knows that there are only two lucky digits — 4 and 7. Also John and Brus decided that digits 3, 5, 6, and 8 are almost lucky because each of them differs from some lucky digit by one.

Some positive integer is called an almost lucky number if it contains only lucky and almost lucky digits in a decimal notation and the number of lucky digits is not less than the number of almost lucky digits. For instance, 7, 48 and 444773 are almost lucky numbers while 485, 404 and 556 are not. You have to find the number of almost lucky numbers between A and B inclusive.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case consists of a single line containing two integers A and B ($1 \le A \le B \le 10^{18}$), separated by a single space.

Output

For each test case print a single line containing the number of almost lucky numbers between A and B inclusive.

Note

The almost lucky numbers between 10 and 100 are 34, 37, 43, 44, 45, 46, 47, 48, 54, 57, 64, 67, 73, 74, 75, 76, 77, 78, 84 and 87.

Example

standard input	standard output
2 1 10 10 100	2 20

Problem E. The Flights (1st Div Only)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	20 seconds
Memory limit:	256 Mebibytes

John and Brus recently bought several small airplanes. And now they want to make some money by holding some flights. There are N cities with airports and it's possible to hold a flight from any city to any other. Somehow Brus found out that the most efficient way is to hold N flights in such way that for each city there is exactly one incoming and one outgoing flight.

John: What is this button for?

Brus: Damn, I've forgotten my parachute!

Some cities are connected by two-way roads. There can be more than one road between two cities. Two cities are called connected if it's possible to get from one city to another by roads (possibly via some other cities). John thinks that it's too silly to hold a flight between two connected cities. Thus he wants to assign all N flights in such way that there will be no flights between connected cities. You have to calculate the number of flight assignments that satisfy John's condition modulo 1234567891.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case starts with a line containing two integers N and M — the number of cities and the number of roads respectively ($1 \le N \le 1000$, $1 \le M \le 10^5$). Each of next M lines contains two different integers A_i and B_i — the cities connected by i-th road ($1 \le A_i, B_i \le N$). All the integers in a single line are separated by single spaces.

Output

For each test case print a single line containing the number of flight assignments satisfying John's condition modulo 1234567891.

Example

standard input	standard output
2 4 3 1 2 3 4 4 3 10 0	4 1334961

Problem F. The Easiest Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Each problemset must contain at least one easy problem. And the easiest problem should be somewhere in the middle. So, here it is right in the middle of the problemset. You can treat it as a gift for you. And here is an advice — just type a few lines of code and submit it as soon as possible. A story for this problem is the following. John and Brus were training for the World Finals. They have solved a lot of the problems at acm.lviv.ua.

John: Hey, Brus, have you solved this problem?

Brus: John, leave me alone — I'm sleeping!

You are given the lists of all John's and Brus's accepted runs and you have to determine X — the number of distinct problems solved by at least one of the guys and Y — the number of distinct problems solved by both of them.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case starts with a line containing two integers N_J and N_B — the number of John's accepted runs and the number of Brus's accepted runs respectively ($1 \le N_J,\ N_B \le 100$). The next line contains N_J integers J_i. Here J_i is the problem number for i-th John's accepted run. The following line contains N_B integers B_i, where B_i is the problem number for i-th Brus's accepted run ($1 \le J_i, B_i \le 10^4$). All the integers in a single line are separated by single spaces.

Output

For each test case print a single line containing two integers X and Y separated by a single space.

Example

standard input	standard output
2 5 3 1001 1002 1001 1006 1099 1007 1007 1099 4 7 1 2 3 4 1 1 1 4 2 2 2	5 1 4 3

Problem G. The Squares

Input file:	`standard input`
Output file:	`standard output`
Time limit:	20 seconds
Memory limit:	256 Mebibytes

John and Brus are practicing multiplying integer numbers. There are N distinct integers written down and each time they choose some subset and try to calculate the product of all the numbers in this subset.

John: And what if we'll choose these numbers?

Brus: And what if we'll not? I'm tired!

Sometimes the result they got is the square of some integer number. You have to calculate the number of nonempty subsets where the product of the numbers is the square of some integer number.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case starts with a line containing the integer N ($1 \le N \le 1000$) — the number of integers written down. The next line contains N integers A_i ($1 \le A_i \le 1000$, all A_i will be distinct), separated by single spaces.

Output

For each test case print a single line containing the number of desired subsets.

Example

standard input	standard output
2 4 3 6 2 1 3 4 9 16	3 7

Problem H. The Fast Reading

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	256 Mebibytes

Brus is trying to develop the fast reading skill. For that purpose he is reading and pronouncing different strings of lowercase letters ('a' — 'z'). It takes him one second to read and pronounce a single letter. But there is a problem with a letter 'r'. Sometimes Brus needs more than a second to read and pronounce this letter. It depends on a letter preceding 'r'.

John: Try this «rarorurereriry».

Brus: Ha-ha, very funny!

John is preparing some strings for Brus to practice. They don't have much time, so John is choosing only nonempty strings that Brus will be able to read and pronounce in N seconds or less. You have to find the number of such strings modulo 1234567891.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case starts with a line containing integer N ($1 \le N \le 10^9$). The next line contains 27 integers A_i separated by single spaces. A_1 is the time needed for Brus to read and pronounce letter 'r' when it appears at the very beginning of the string, A_2 — when 'r' is preceded by letter 'a', A_3 — by 'b' and so on ($1 \le A_i \le 10$).

Output

For each test case print a single line containing the number of desired strings modulo 1234567891.

Example

standard input
2
1
2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
4
3 1 1 9 1 3 1 1 1 7 5 8 7 1 7 9 7 9 7 7 10 6 3 1 3 9 6

standard output
25
422401

Problem I. The Best Picture

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	256 Mebibytes

Every winter John and Brus play curling on the street. There is nobody else who can play this funny game, so they have to play against each other. On the other hand there are a lot of spectators, especially young ladies, who are interested not only in this winter sport.

John: I'm not sure whether she is looking at me or you!

Brus: But I'm sure you are staring at her now.

This time guys decide to set up a competition. They will play one game each of next N days. If John wins a single game he will get a medal from his female fan club, otherwise he will return all the medals he has to the club. After each game a picture of John, Brus and their fans will be taken. Of course, John will wear all the medals he has. After the competition is over John will choose the best of N pictures he has (i.e. the picture with the maximal number of medals) for his favorite social network. The probability that John will win a single game is P. You have to determine the expected number of medals in the best picture John will choose. Note that initially John has no medals.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case consists of a single line containing N and P ($1 \le N \le 1000$), ($0 \le P \le 1$), separated by a single space. P will be given with exactly four digits after the decimal point.

Output

For each test case print a single line containing the expected number of medals in the picture. Your answers must be rounded (using standard rounding rules) to seven digits after the decimal point.

Example

standard input	standard output
2 3 0.5000 1 0.4774	1.3750000 0.4774000

Problem J. The Flowers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	16 seconds
Memory limit:	256 Mebibytes

John and Brus often visit a forest not far from their home. There are N beautiful glades and $N-1$ two-way footpaths in the forest. Each footpath connects two glades. Also it's possible to get from any glade to any other using footpaths. Thus there is exactly one simple path (i.e. path that consists of distinct glades) from one glade to another.

John: Brus, these flowers are for you!

Brus: Nice, they are beautiful! But why there are four of them?

Guys will visit the forest for M successive days. Before the first day there are F_i very beautiful flowers growing on i-th glade. At the beginning of each day one additional flower will appear on each glade. After that John and Brus will either pick all the flowers on a glade or calculate the number of flowers on the simple path between two glades inclusive.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case starts with a line containing integers N and M ($1 \le N, M \le 10^5$). The following line contains N integers F_i ($0 \le F_i \le 10^9$). Each of next $N-1$ lines contains integers A_i and B_i — the glades connected by i-th footpath. Each of next M lines starts with a single character ('P' or 'C'). If it is 'P' then it will be followed by integer G_i and it means that on i-th day guys will pick all the flowers on glade G_i. Otherwise 'C' character will be followed by integers U_i and V_i. In this case John and Brus will calculate the number of flowers on a simple path between glades U_i and V_i inclusive ($1 \le A_i, B_i, G_i, U_i, V_i \le N$). All the integers in a single line are separated by single spaces.

Output

For each test case and for each calculation print a single line containing desired number of flowers.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>2
2 3
4 7
1 2
C 1 2
P 2
C 1 2
1 1
100
C 1 1</pre></td><td><pre>13
8
101</pre></td></tr>
</table>

Problem K. The Card Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

In the evening John and Brus play very funny card game. It is played with 16 unique playing cards in four suits (Spades, Clubs, Diamonds and Hearts) and four ranks (Jack, Queen, King and Ace. Here ranks are listed in increasing order). We will denote the suit and the rank with a single uppercase letter — 'S' (Spades), 'C' (Clubs), 'D' (Diamonds), 'H' (Hearts), 'J' (Jack), 'Q' (Queen), 'K' (King) or 'A' (Ace). Thus each card can be represented as a two-character string where the first character is a card rank and the second — a card suite.

John: Can't find any joker!

Brus: Maybe, it's because I got them all!

At the beginning of the game cards are shuffled and each of two players gets eight cards. There is a main deck with cards faced up which is initially empty.

The two players alternate turns. At each turn player either places onto the top of the main deck one of his cards with a rank greater than or equal to the rank of the card that is currently on the top (or any card if the deck is empty) or takes three cards from the main deck (or all the cards if the main deck contains less than three cards). If after a turn a player has no cards he is considered to be a winner.

Both players use an optimal game strategy. If the player can win, he will follow the strategy that minimizes the number of moves in the game. If the player can't win, he will follow the strategy that maximizes the number of moves in the game. You have to determine the game outcome.

Input

The first line contains single integer T ($1 \le T \le 47$) — the number of test cases. Each test case consists of a single line containing eight cards C_i John has at the beginning of the game. All the cards will be separated by single spaces. The next line contains the name of the player (either John or Brus) who will start the game. All C_i will be distinct.

Output

For each test case print a single line containing the name of the winner and the total number of moves in the game separated by a single space (or `Draw` if the game is endless).

Example

standard input	standard output
2 AS AC AD AH KS KC KD KH Brus JS JC QD QH KS KC AD AH John	John 16 Draw

Problem L. The Balloons (2nd Div Only)

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 Mebibytes

John and Brus are playing with colored balloons. John has N balloons and Brus has M balloons.

John: What is your favorite color?

Brus: The first one, of course!

Guys would like to achieve $A : B$ balloon relation. For this purpose they will repeat the following action. If $N \times B$ is greater than $M \times A$ then John will blow up one of his balloons and if $N \times B$ is less than $M \times A$ Brus will blow up one of his balloons. They will stop if there is no balloons left or if $A : B$ relation is achieved. You have to find the number of John's balloons and the number of Brus's balloons at the end of the day.

Input

The first line contains single integer T ($1 \leq T \leq 47$) — the number of test cases. Each test case consists of a single line containing four integers N, M, A and B in this very order ($1 \leq N, M, A, B \leq 10^9$). All the integers are separated by single spaces.

Output

For each test case print a single line containing the number of John's balloons and the number of Brus's balloons separated by a single space.

Example

standard input	standard output
2 5 8 2 3	4 6 0 0 1 4 10 35