Problem A. Albert's Challenge

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

Albert E. loved math riddles, he used to give his students a new math challenge at the end of each lecture and it would keep them busy for weeks.

One of his challenges goes as follows:

Given a sequence of integers, can you find the longest non-decreasing non prime sub-sequence?

Input

The first line of the input contains an integer T $(1 \leq T \leq 128)$, the number of test cases.

The first line of each test case contains a single integer N $(1 \leq N \leq 10^3)$ the length of the array. The second line contains N space separated numbers a_i $(1 \leq a_i \leq 10^7)$.

Output

For each test case print a single integer, the length of the longest non-decreasing non prime subsequence.

Examples

albert.in	Standard Output

1
9
6 4 5 4 3 4 8 3 9
	5

Problem B. Bohemian Rhapsody

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

Donald K. loves music, he can't even write code without background music to inspire him, his favorite all-time-band is Queen, he listened to their entire discography so many times, and he wanted to use his brilliant coding skills to create a playlist from their songs.

He labeled each song with a specific integer, and made two playlists, one of them contained the songs labeled with sorted odd integers, and the other contained the songs labeled with sorted even integers.

Given an unsorted array of numbers of songs, can you print the two playlists' numbers?

Input

The first line of the input contains an integer T $(1 \leq T \leq 1024)$, the number of test cases.

The first line of each test case contains a single integer N $(1 \leq N \leq 10^3)$ the number of songs. The next line contains N space separated integers A_i $(1 \leq A_i \leq 10^3)$.

Output

For each test case print N space separated integers, the sorted odd numbers the the sorted even numbers.

Examples

donald.in	Standard Output

1
9
9 8 7 6 5 4 3 2 1
	1 3 5 7 9 2 4 6 8

Problem C. Comer Test

Program:	`comer.(cpp|java)`
Input:	`comer.in`
Balloon Color:	`Silver`

Prof. James Comer is known for his famous difficult tests, in fact, his students invented the term "Comer Test"(named after his last name) to refer to a very difficult exam. James loves to integrate games and puzzles with his tests, so one day, he gave his students the following board game:

You are given a grid that we will represent in the following form:

- '.' empty cell, can pass.
- '*' wall, can't pass.
- cells with doors labeled from $[a - z]$, when you reach a door you are free to choose between two options, either treat it like an empty cell and move on with the game, or move to another cell with a door of the same letter or a letter that comes after it in the alphabet (if it exists in the board). For example: if you are in a cell with letter a, you are only allowed to move to cell with letter a or cell with letter b (if exists) or treat it as an empty cell.
- 'S' represents the start and 'T' represents the target.

Game rules:

- With a cost of 0, you are allowed to use the door feature several times in the whole game.
- With a cost of 1, you can pass from a cell to any of the four adjacent cells (up, down, left, right) if and only if it doesn't contain a wall.

James challenged his students to figure out the minimum cost required to get from the start S to the target T, are you up to that challenge?

Input

The first line of the input contains an integer T ($1 \leq T \leq 1024$), the number of test cases.

The first line of each test case contains two integers r and c ($1 \leq r, c \leq 100$) and r X $c \geq 2$. Then r lines follow, each line contains c characters and each character will be one of { '.', '*', lowercase english letter [a-z], 'S', 'T' }.

It's guaranteed that each grid will contain exactly one start S and exactly one target T.

Output

For each test case if it's possible to reach T from S print a single integer, the minimum cost required to get from S to T, otherwise print -1.

Examples

comer.in	Standard Output
1 7 11 *********** *S.a....d.* *...b.....* *ccc***ccc* *d........* *..a....bT* ***********	3

Problem D. Die or Win

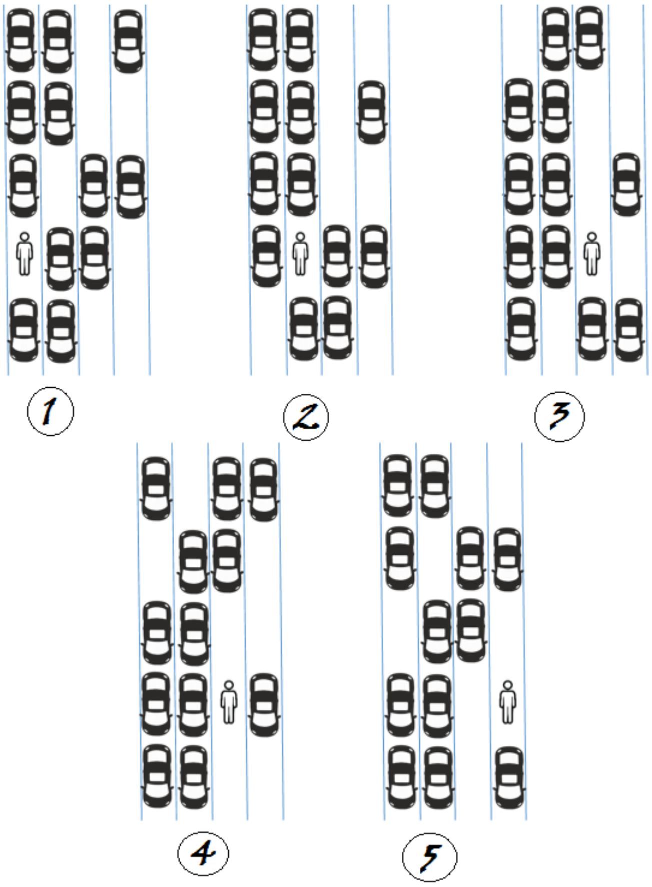
Program:	`die.(cpp|java)`
Input:	`die.in`
Balloon Color:	`Black`

Alan T. and Ada L. loved extreme games and dangerous adventures, and they wanted to invent their own extreme game that no one thought of before.

There was a wide highway close to their house, it consisted of 4 lanes, what they would do is that they would assign K units of time and challenge each other if they can manage to cross the street without getting killed! However, they figured that since it's a "deadly" game, it would be a good idea for them to use their good math skills to predict if they would win or die.

They studied the streets traffic nearby, and managed to represent each lane at every second with 4 digits of zeros and ones, where one represents a car existing in that lane in the given moment in time, and zero represents no car.

Given a segment of the street of length 10, represented in the previously mentioned way, and considering that a player has two options, in each second (s)he can move forward in a straight line to the next lane or remain in the same position, can you figure out if he would make it to the other side or not? and if he could, can you know the minimum number of seconds that would take him to reach it?



The above picture illustrates a car moving and Sam walking through.

Alan here should start from the left at the position 3 counting from 0 and the walking instructions here for each column will be : 1 1 2 1 where the number in the circle is the number of second when Sam was on that lane, and in this example he stayed for a total of 5 seconds on the road.

Input

The first line of the input contains T $(1 \leq T \leq 1024)$ the number of test cases.

The first line of each test case contains a single integer K $(1 \leq K \leq 10)$ the initial position of the player. Then 10 lines follow, each line contains a string of zeros and ones representing the current 4 lanes.

Output

For each test case, if there is a winner, print the minimum number of second to cross the street, otherwise print -1.

Examples

die.in	Standard Output
1 10 0000 0110 1010 1101 1010 1010 1000 1001 1000 0100	4

Problem E. Edison's Tree

Program:	`edison.(cpp|java)`
Input:	`edison.in`
Balloon Color:	`Purple`

Thomas E. once said, "I have not failed, I've just found 10000 ways that won't work", and he said that based on his own life, since he used to experiment a lot, he found it hard to follow up with his successful and unsuccessful experiments, so he asked his friend Alan T., the computer scientist to help him keep track of his experiments.

Alan drew a tree, that represents Thomas's experiments' outcomes throughout his life, each node has a value of 0 (unsuccessful experiment) or 1 (successful experiment).

Thomas loved the idea, because it was so easy for him to find out the number of successful experiments starting from any point in time.

Given a directed tree, can you help him find out how many 1s are there in every subtree?

Input

The first line of the input contains an integer T $(1 \leq T \leq 1024)$ the number of test cases.

The first line of each test case contains a single integer N $(1 \leq N \leq 10^3)$ the number of experiments. The second line contains N space separated integers $A_i \in \{0, 1\}$ where 0 represents unsuccessful experiment and 1 represents successful experiment. Then $N - 1$ lines follow, each line contains 2 integers u and v $(1 \leq u, v \leq N)$, represents a directed edge from u to v.

Output

For each test case print N space separated integers, the i_{th} integer represents the number of ones in its subtree.

Examples

edison.in	Standard Output

1
5
0 1 0 1 1
3 1
3 2
2 4
1 5
	1 2 3 1 1

Problem F. Finding New Elements

Program:	`elements.(cpp|java)`
Input:	`elements.in`
Balloon Color:	`Gold`

Marie C. works in her lab every day on discovering new elements, to keep track of her work, she assigns small letters from a to z to each outcome of her experiments, and writes down the sequence of outcomes at the end of the day as a string. She considers it good luck whenever she has identical outcomes in a row, and she calls it a good string.

For example *aaaaaa* is a good string and aab is not.

Help Marie evaluate her day at the lab by answering two types of queries:

- *type* 0: given position i and char C, Change the character in the i_{th} position into a character C.
- *type* 1: given position L and R, print length of the longest good substring between L and R inclusive.

Input

The first line of each test case contains T $(1 \leq T \leq 128)$ the number of test cases.

The first line of each test case contains a string S $(1 \leq |S| \leq 10^5)$. The second line contains a single integer Q $(1 \leq Q \leq 10^5)$ the number of queries. Then Q lines follow, each line contains more than one integer, the first integer *type* $\in \{0, 1\}$:

- if *type* equals 0, an integer i $(1 \leq i \leq |S|)$ and small english letter C.
- if *type* equals 1, two integers follow, L and R $(1 \leq L \leq R \leq |S|)$.

$|S|$ represents the length of the string S.

Output

For each test case, for each query of *type* 1, print a line containing a single integer, the length of the longest good substring.

Examples

elements.in	Standard Output
1	2
hossamyousef	4
7	3
1 3 8	
0 5 o	
0 6 o	
0 7 o	
0 2 c	
1 3 8	
1 6 11	

Problem G. Game of Nine

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

Sheldon C. has a strange OCD, he only feels comfortable dealing with numbers that the sum of its digits equals 9, and he called those numbers, the chain of 9.

He used to play a game with his friends based on his disorder, he would give them a number X that belongs to the chain of 9, and another number K, the winner would be the person that calculates the next K_{th} number that belongs to the chain of 9. For example, if $X = 9$ and $K = 2$, the answer would be 27. Given X and K, can you win this game?

Input

The first line of input is the number of test cases T $(0 \le T \le 5000)$, then coming T test cases, each test case in a line and has two numbers separated by a space.

The first number x $(0 < x < 10^{15})$ represents a number belongs to the chain of 9.

The second one y $(0 \le y \le 10^{15})$ represents the number of counts.

Output

For each test case, print a line containing the number in the chain of 9 after y counts from the number x where the sum of its digits equals 9.

Examples

game.in	Standard Output

3
9 2
63 5
54 3
	27
	108
	81

Problem H. Houston, We Have A Problem

Program:	`houston.(cpp|java)`
Input:	`houston.in`
Balloon Color:	`Yellow`

Astronaut Jim L., decided to cryogenically freeze himself and wake up 1000 years in the future! When he woke up in the year 3015, interstellar travel was common and he decided to take a tour to a few neighbor galaxies. As he flew his interstellar spaceship through the stars he got stuck among a group of black holes. The center of black hole number i ($i > 0$) exists at point (x_i, y_i, z_i).

He needed to navigate very carefully to avoid getting trapped within the enormous gravitational force. He had to find a particular place in space where he could minimize the total effect of the black holes but to do this he had to do a few calculations: If Chris was placed at some point (x_0, y_0, z_0) then the effect of the black hole number i on him is the squared distance between him and that black hole, which is given in the following formula:

$(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2$

Thus, the total effect is:

$$\sum_{i=1}^{n} (x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2$$

Your task is to place Chris's spaceship in a certain point in space where the total effect of the black holes is at the minimum.

Input

The first line of the input contains one integer T ($1 \leq$ T ≤ 50) representing the number of test cases. Next lines will contain the test cases: each test case starts with N ($1 \leq N \leq 10^5$) indicating the number of black holes. Line number i ($1 \leq$ i $\leq$ N) consists of three space separated integers: x_i, y_i, z_i respectively, ($-10^6 \leq x_i, y_i, z_i \leq +10^6$), representing the coordinates of the center of the black hole number i.
Warning: large input data files.

Output

For each test case you have to print three floating-point numbers each in line by itself to 4 decimal places representing x_0, y_0, z_0 respectively. The output represents the coordinates in which we should place our hero in order to minimize the effect of black holes on him.

Examples

houston.in	Standard Output
3	1.5000
2	1.5000
1 1 1	1.5000
2 2 2	0.3333
3	0.3333
0 0 1	0.3333
0 1 0	3.0000
1 0 0	6.8000
5	12.0000
2 5 13	
8 21 2	
1 5 8	
3 1 34	
1 2 3	

Problem I. Isaac’s Triangles

Program:	`issac.(cpp|java)`
Input:	`issac.in`
Balloon Color:	`Blue`

Isaac N. was messing around in Geogebra (mathematics software that can be used in drawing geometrical shapes) one day. As he was staring in the grid on his screen, he wondered to himself, how many triangles can be drawn using N points?

Given N points ($3 \leq N \leq 1000$), can you figure out how many triangles can be formed using them?

Input

You will be given T ($1 \leq T \leq 128$) number of test cases. Each of the following T lines will contain an integer number N ($1 \leq N \leq 1000$) which denotes the number of points that can be used to form a triangle. You can assume that **no three points are on the same line**.

Output

For each test case print in a single line a number that indicates the number of triangles which can be formed using N points.

Examples

issac.in	Standard Output
2 5 1000	10 166167000

Problem J. Just Another Space Problem

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

Carl S. was once studying the chemical compounds of asteroids that hit earth since the beginning of the universe, or to be more precise, whatever asteroids he managed to get his hands on.

He wanted to keep it simple and optimize his study, so he, somehow, represented each asteroid with an integer number, and wrote all these numbers in a list.

He would know that a pair of asteroids is from the same planet formation by permuting the two integers, then comparing them after removing any leading zeros (zeroes on the left of the number which can also be ignored) that might come up. If at any moment the two permutations are equal, he would know that this pair of asteroids originated from the same planet formation long long time ago.

Given a list of numbers, can you find out how many pairs of similar asteroids are there?

Input

You will be given T ($1 \leq T \leq 128$) number of test cases. For each test case you will be given an integer number N ($1 \leq N \leq 1000$) which denotes the number of asteroids, then follow N numbers a_0, a_1, ... a_{N-1} ($1 \leq a_i \leq 10^{12}$) separated by blank spaces representing each asteroid.

Output

For each test case, print in a single line an integer representing how many pairs of similar asteroids exist.

Examples

space.in	Standard Output

3
3
10 1 100
2
17 27
6
1 2 21 120 20 10
	3
	0
	3

Problem K. Kepler's Laws

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

Johannes K. is known to be brilliantly smart, but unfortunately he had a very poor memory, his biggest problem was remembering names, so he created a method that helps him distinguish the students helping with his research without having to memorize their real names.

He gave each field of research he is working on a certain string, and gave the researchers working under this field a subsequence of the field's name as a nickname, this way he knew where everyone works, and also gave them unique nicknames that are easy to memorize!

Given 2 strings, field name and researcher name (not necessarily in this order), can you check if one of them is a subsequence of the second or not?

Input

The first line of each test case contains T $(1 \leq T \leq 1024)$ the number of test cases.

Each test case contains 2 strings, each string S $(1 \leq |S| \leq 10^3)$ will be given in a single.

|S| represents the length of string S.

S contains only uppercase english letters.

Output

For each test case print "YES" if at least one of them is a subsequence of the other string, otherwise print "NO".

Examples

kepler.in	Standard Output
3 WEYYDI DTLCOUEGM FWURP WUP HTQVKG BHTYTSZOT	NO YES NO