競賽開始前請勿翻頁。

Problem A. Hungry Lacy

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

Winner Winner Chicken Dinner!

Lacy is always very hungry after he win the PUBG (PROGRAMMER'S UNKNOWN BUGGROUND) champion. So he has a refrigerator to store many cookies.

There are 2 favors of cookies. One is A and the other is B Cookies. Lacy will eat some cookies in interesting regular after every time he won the PUBG game. Let A_n, B_n mean how many A and B cookies did Lacy eat after he won the n-th round PUBG game. It is exist that

- $A_n = wB_{n-1} + xA_{n-2}$
- $B_n = yA_{n-1} + zB_{n-1}$

Giving $A_1, A_2, B_2, w, x, y, z$, Can you calculator how many cookies totally did Lacy eat from he won the first round to n-th rounds PUBG game? Because the answer maybe very large, just output the answer after mod $10^9 + 7$.

Input

The first line of the input contains an integer `t` which is the number of test cases. For each test there are 8 integer `N a1 a2 b1 w x y z` in a line. The meaning are as same as problem description

Output

For each testdata output the answer in a line.

Specification

$1 \leq t \leq 10^3$

$0 < N \leq 2^{63} - 1$

$0 \leq A_1, A_2, B_1, w, x, y, z \leq 10^9$

Examples

standard input	standard output
2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1	2 5

This page is intentionally left blank

Problem B. Pusheens

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds

There are N *pusheens* living in NTU, and each of them has its weight w_i and *jumping ability* h_i.

They are now piled up from bottom to top. To make the stacking physically stabler, they want to adjust the stacking so that they are "sorted" by weight, which means the heaviest pusheen is at the bottom, the second heaviest is just upon it, ..., and the lightest is on the top. The steps they can take are of the following two kinds:

- One pusheen in any pile jumps down to the ground and forms a new pile with all the pusheens on his back.
- The pusheen at the bottom of any pile, carrying the whole pile, jumps onto the top of another pile.

Note that in either of these steps, all pusheens on the jumping pusheen moves with him, so the jumping pusheen's jumping ability must be no less than the total weight it carries (not including his own weight).

Given the initial stacking and every pusheen's weight and jumping ability, please find out the least number of steps the pusheens have to take in order to reach their goal.

Input

The first line is an integer N, the number of pusheens living in NTU. Among following N lines, the i^{th} line is two integers w_i and h_i, indicating the i^{th} pusheen's weight and jumping ability respectively. The final line is N integers $a_1, a_2, ..., a_N$, indicating the index of pusheens from top to bottom in the initial stacking.

The followings are guaranteed:

- $1 \leq N \leq 1 \times 10^6$
- $1 \leq w_i \leq N$ and w_i are all distinct
- $0 \leq h_i \leq 10^{18}$

Output

Output a single integer: the least number of steps pusheens have to take to achieve their goal.

Examples

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

Explanations

In the first sample, the optimal solution is as follows:

1. The heaviest pusheen jumps down.
2. The lightest pusheen jumps down.
3. The pusheen with weight 2 jumps onto the heaviest one.
4. The lightest pusheen jumps onto the one with weight 2.

Problem C. Vanilla Ice

Input file:	standard input
Output file:	standard output
Time limit:	1 second

Vanilla Ice is the most loyal man of Dio. His stand - Cream, can send everything in a sphere to the darkness space. Creating a sphere with a radius of r will require r^2 of energy, so if the radius of the sphere is too big he will become very tired.

Today there are n Aswan Houseflies appear in Dio's bedroom. Vanilla Ice wants to use his stand to send these flies into the darkness space. He can use his stand Cream in any place in the room but only once, How much energy does he need to consume at least to wipe out all the flies? We can assume that Dio's room is in Euclidean space and each fly is a point in space. If Vanilla Ice's sphere can contain all the points in space, it can eliminate all flies.

Input

The input consists of multiple data sets.

The first line of a data set contains an integer n, which is the number of flies. It satisfies the condition $1 \leq n \leq 1000$.

The location of n flies are given by three-dimensional orthogonal coordinates: $(x_i, y_i, z_i)(i = 1, ..., n)$. Three coordinates of a fly appear in a line, separated by a space character. Each value is given by a decimal fraction, and is between -10000 and 10000 (both ends inclusive). Points are at least 0.01 distant from each other.

The end of the input is indicated by a line containing a zero.

Output

For each data set, the square of the radius of the smallest sphere containing all given points should be printed, each in a separate line. The printed values should have 5 digits after the decimal point.

Examples

standard input	standard output
4 0.00000 0.00000 0.00000 0.01111 0.00000 0.00000 0.02222 0.00000 0.00000 0.03333 0.00000 0.00000 5 5 0 0 -5 0 0 0 3 4 4 -3 0 2 2 -2 1 7122.0 -7122.0 7122.0 0	0.00028 25.00000 0.00000

This page is intentionally left blank

Problem D. Utopia

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

Albert the philosopher thinks that inequality of wealth is the source of all problems in the world. In the **utopia** in his mind, the most important thing is **equality**, that is, every person's annual income and outcome should be equal each year.

As an experienced system developer, Albert has built a system that can record all transactions in the country. One year after the system has been published, Albert is now broadcasting the summary of all recorded transactions to the public, so that everyone can see whether they have reached **equality** or not.

To add excitement to the broadcast, Albert does it this way:

1. All transactions are listed in the order of timestamp, but initially only the two participants of every transaction is visible (which one the sender/receiver is and the transferred amount are hidden).
2. The hidden information for each transaction is revealed one by one in the order of timestamp.

However, people nowadays are quite impatient: once the revealed information already tells them whether or not the **equality** is reached, they will quit the broadcast immediately.

As an assistant of Albert, you already know all the informations. Assuming that people in Albert's country are all **extremely smart**, please find out when will people leave the broadcast.

Input

The first line is an integer T, indicating the number of transactions this year.

Each of the following T lines are three integers S_i, R_i and V_i, indicating the sender id, receiver id and the transferred amount of the i^{th} transaction. The lines are given in the order of timestamp. The followings are guaranteed:

- $1 \leq T \leq 10^6$
- $1 \leq S_i, R_i \leq 2 \times 10^5$
- $0 \leq V_i \leq 10^9$

Output

Print a single integer k, meaning that people leave exactly at the revealation of the k^{th} transaction.

Examples

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

Explanation

In the first sample, equality is reached, so the result is unknown until the last transaction is revealed.

In the second sample, the person with id 1 pays one dollar on the first transaction and participate in no transactions afterward, so it'll be known at the beginning that equality is not reached.

Note that the transferred amount of a transaction may be zero, which means that the transaction does not involve money (exchange of stock and real estate, for example).

Problem E. Annie in Wonderland

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds

Annie is a curious girl, live with her lovely animal friends in a small wooden house in the forest.

One day, she got very tired of sitting on the bank, and of having nothing to do. Suddenly, she saw a weird white rabbit with pink eye running in her sight. The rabbit looks flurried and just murmuring something. It ran away immediately. Annie was very inquisitive about where the rabbit goes, she stalked the rabbit carefully. However, she was too concentrated on the rabbit, so she neglected a big hole on the ground and fell into the hole.

After poped down the hole, she found that there is actually a underground world created by the rabbits. There are n chambers in the underground world, and there are $n-1$ bidirectional corridors, each connecting a different pair of distinct chambers. It is guaranteed that for each pair of the chambers, there must be a path connecting each other.

In each chamber, there is a tree with a magic shelter around it, the tree produces some flavor of wafer. Because of the magic shelter, you can only take one wafer from the tree every time you get into the chamber.

Annie loves the desserts, of course she likes the wafers, and she want to know how much happiness she can get from eating every kind of wafers while every kind of wafers provide k_i value of happiness to Annie, and her happiness will be the maximum value of happiness she had got. But Annie is lazy, she must walk along the shortest path every time she walk from the chamber a_j to the chamber b_j. She will eat the wafers every time she goes through a chambers.

Now, a_j and b_j is given, your task is to help Annie to know how much happeness she can get from taking all the wafers on the path.

Input

The first line of input contains an integer t which is the number of test cases. For each test case, the first line contains an integer n, representing the number of the chambers. Then there are $n-1$ lines, each contains two different integer u_i and v_i, which indicating that there is a bidirectional corridor between the ${u_i}^{th}$ chamber to the ${v_i}^{th}$ chamber. Then there is one line contains n integers k_i, representing the value of happiness the i^{th} wafer would provides. And the next line is an integer m. Then there is m lines, each contains a pair of integers a_j and b_j, indicating that Annie will walk from the ${a_j}^{th}$ chamber to the ${b_j}^{th}$ chamber.

Output

For each (a_j, b_j), output the maximum happeness Annie can get from taking all the wafers on the path.

Specification

- $1 \leq t \leq 10$
- $1 \leq n, m \leq 10^5$
- $1 \leq u_i, v_i, a_j, b_j \leq n$, where $i \in [1, n]$ and $j \in [1, m]$
- $|k_i| \leq 10^9$, where $i \in [1, n]$

Examples

standard input	standard output
1 12 1 2 2 3 3 4 3 5 2 6 6 7 7 8 6 9 6 10 10 11 11 12 10 4 6 6 6 5 4 5 4 4 5 10 4 1 12 3 9 7 11 8 9	10 6 5 5

Problem F. Ponchi want QuantumSuit

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second

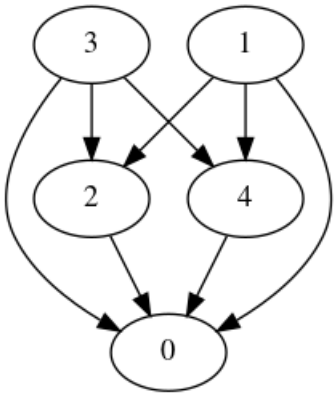
Ponchi is a minecraft player. He collect resources every day because he want Quantumsuit. Let's help him to get Quantumsuit.

Quantumsuit must need some components to create,for example, shoes,armors and shirt.Every components can make parallelly,and all of Quantumsuit components may need some extra components to craft. Hence,if you want to build Quantumsuit , you need to build the basic components first.

And Making components needs time to craft. Please calculate the total time it takes to create all components. Every components only has to be created once even if it is prerequisite for several components. (The crafting will **not** change).

There are no cicular dependency. Hence,everything can be maked.

Following picture show the sample input dependency graph.



Input

The input begins with single integer N(0 < N <= 100)indicates total amount of components(include Quantumsuit).

Follow N line format like K COST L M1 M2 .... ML indicates K (K is the number between 0 to N-1) components need COST (integer,0 < COST < 1000)time to build.And if you need to craft this components,you need to collect M1 M2 ... ML (0 <= Mi <= $N-1$)components first. Every components is necessary. The total time is all components finish.

Output

Please output total time in one line.

Examples

standard input	standard output
5 0 199 4 1 2 3 4 1 935 0 2 879 2 1 3 3 522 0 4 146 2 1 3	2013

This page is intentionally left blank

Problem G. Exponentiation

Input file:	`standard input`
Output file:	`standard output`
Time limit:	30 seconds

Give you an interger n, please print out n^{71222}

Input

The input consists of multiple data sets.

The first line of a data set contains an integer T, which is the number of data sets. It satisfies the condition $1 \leq n \leq 21$.

The first line of each data set contains an integer n. It satisfies the condition $-10 \leq n \leq 10$.

Output

For each data set, n^{71222} should be printed, each in a separate line.

Examples

standard input	standard output
3 -1 0 1	1 0 1

This page is intentionally left blank

Problem H. Picky's Worry

Input file:	`standard input`
Output file:	`standard output`
Time limit:	12 seconds

Picky is an alien from a planet far away from the earth. He just came to the earth, and found that the buildings here are too ugly in the aesthetic of the aliens.

In Picky's country, they use an integer to represent how beautiful a buildings is, which calls "gorgeousity". The building have larger "gorgeousity" is more beautiful. There is a "Beautify Laser Gun" on Picky's UFO. If he shoot the laser beam on some buildings in a row whose "gorgeousity" is $x_l, x_{l+1}, ..., x_r$, add $\alpha k^2 + \beta k + \gamma$ to the $l + k^{th}$ buildings.

Picky hired you to write a program to simulate what will happen if Picky use his laser gun on a street, and calculate the sum of the "gorgeousity" mod $10^9 + 7$ of some buildings.

Input

The first line of the input contains an integer t which is the number of test cases. For each test case, the first line contains an integer n, representing the number of the buildings on the street, the second line contains three integers α, β, γ, representing the parameters of the alien's laser gun, the third line contains n integers x_i, the "gorgeousity" of each building, the forth line contains the number of operations m. There are m lines in the last part. Each of them is either '1 l r' (shooting beautify laser beam to the $l^{th}, l + 1^{th}, ..., r^{th}$ buildings) or '2 l r' (output the sum of "gorgeousity" of the $l^{th}, l + 1^{th}, ..., r^{th}$ buildings mod $10^9 + 7$).

Output

For each operation of the second kind, write their output on the separate line, in order they appear in the input.

Specification

- $1 \leq t \leq 5$
- $1 \leq n, m \leq 10^5$
- $-100 \leq \alpha, \beta, \gamma \leq 100$
- $0 \leq x_i < 2^{31}$, where $i \in [1, n]$
- $1 \leq l, r \leq n$

Examples

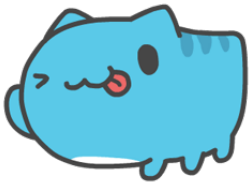
standard input	standard output
1 3 1 2 3 17 239 999 4 2 1 3 1 2 3 2 2 3 2 1 2	1255 1247 259

This page is intentionally left blank

Problem I. Capoo Loves Mahjong

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second

Bugcat Capoo is a well-known character on the Internet. He eats anything you can imagine. He can even eat his friend, DogDog, for instance. If you don't know who is Capoo, his cute picture is attached below.



However, Capoo and his friends have a deep secret that nobody knows. They considerably immerse themselves in the Chinese classical and traditional game - Mahjong. On every Thursday night, they will gather together at Capoo's home, sit around a square table, and compete all night long. They will play at least 4 matches(將) on each night. Consequently, when it reaches the final match, all of them will feel extremely exhausted and therefore unable to concentrate on the Mahjong tiles. As it approaches to the end, they can not even decide whether they can yell out "Mahjong!" or not. (Yelling out "Mahjong" means the player has won the game.) Accordingly, they need a Mahjong genius to design a program for them to decide if there hand tiles are winning or not.

However, most of you are not familiar with Capoo Mahjong. To make you a Mahjong genius and solve the problem for Capoo, the followings are the rules of Capoo Mahjong, which is a variation of traditional Taiwanese/Cantonese/Japanese Mahjong.

There are three types of tiles in Capoo Mahjong: Simples Tiles, Honors Tiles, Bonus Tiles.

- Simples Tiles:

  Consists of three suits of simples tiles: Wan(W), Bing(B), Tiao(T). In each suit, the tiles are numbered from 1 to 9. For example, Five Bing(5B) and Three Tiao(3T) are both Simples Tiles. There are 4 identical copies of each Simples Tiles.

- Honors Tiles:

  There are two different sets of Honors tiles: Winds and Dragons. The Winds are East(EE), South(SS), West(WW) and North(NN). The Dragons are Red Center(RC), Green Fortune(GF) and White Board(WB). These tiles have no numerical sequence like the simples. There are 4 identical copies of each Honors Tiles.

- Bonus Tiles:

  There are two sets of bonus tiles in Taiwanese Mahjong: Flowers and Seasons. However, Capoo and his friends doesn't use this kind of tiles since it is a little bit complex. As a result, Capoo Mahjong has no bonus tiles.

A winning hand consists of 17 Mahjong tiles, including exactly one pair of "eyes," and exactly five "sets".

- Eyes means a pair of identical Mahjong tiles. You may form a pair of eyes with either Simples Tiles or Honors Tiles.
- A set can be one of the following two forms,
  - Pong
    Pong is formed by a set of three identical Mahjong tiles. You may form a Pong with either Simples Tiles or Honors Tiles.

– Chow

Chow is formed by a set of three Simples Tiles in the same suit and their numbers are absolutely sequential. For example, 1W 2W 3W forms a Chow Set, but 9B 1B 2B does not.

Please help Capoo and his friends to decide whether a set of 17 Mahjong tiles forms a winning hand!

Input

The first line of the input file contains an integer T, denoting that there will be T testcases in the file.

For each testcase, there will be exactly two lines.

The first line contains a string **N** indicating the player's name.

The second line contains 17 2-character strings, denoting the 17 hand tiles of a player. The tile strings are **NOT** guaranteed to be sorted.

Constraints

$1 <= T <= 50$

$1 <= |N| <= 20$

All the tile strings belongs to the following set: {[1-9][WBT], EE, SS, WW, NN, RC, GF, WB}

It is guaranteed that no more than 4 identical tiles will appear in the hand tiles simultaneously.

Output

For each testcase, if the player's hand tiles is winning and he can yell out "Mahjong", please output **N** can yell out Mahjong!"(without quote), where " **N** is the player's name. Otherwise, please output "Nothing happened."(without quote)

Examples

standard input
2 Capoo 1W 1W 1W 2W 2W 4W 4W 4W 3T 4T 5T 7T 8T 9T GF GF GF DogDog 1B 2B 2B 2B 3B 3B 3B 4B 4B 1T 1T 1T 1T WW WW WB WB
standard output
Capoo can yell out Mahjong! Nothing happened.

Note

Thanks to Wikipedia for offering lots of Mahjong proper noun translation.