A. Absolutely Lucky Controller

Egor works as a controller in the bus. Each day he is given a pack of tickets which he then sells. Recently he has become interested about how many tickets in the pack are lucky. He thinks that the more tickets are lucky the luckier day he will have. Now he wants to find out how lucky for him the next day is going to be. Each ticket number consist of n digits. The ticket is considered to be lucky if the sum of the first $n/2$ digits equals to the sum of the last $n/2$ digits. Egor knows that the numbers in the pack that he will be given can start with equal probability from any number in the interval from a to b inclusively. The pack holds k tickets. The numbers of the tickets are consecutive. Help Egor to find an expected quantity of lucky tickets in the pack.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

Input consists of a single line with three integers a, b, k ($0 \le a \le b < 10^{12}, 1 \le k \le 100000$). Integers a and b consist of same amount of digits, and this amount equals to the amount of digits in the number of each ticket. They may start with zeroes. The amount of digits in a and b is always even.

Output

Output the expected quantity of lucky tickets in the pack in the form of irreducible fraction. In case the result is an integer – no slash should appear in the output.

Sample Input

```
3
0123 4567 150
10 10 20
4000 4999 11
```

Sample Output

```
6519/635
2
103/125
```

(This page is almost empty.)

B. Best Piano Chords

Little Zhik is a celebrated pianist, who received a super piano from Dr.Ckyly recently. Little Zhik would like to play the pleasantest chord on this super piano.

There are n keys on the super piano so one can play n different single notes, numbered from 1 to n. When i-th single note is played, it associates a *pleasant value* A_i.

A *super chord* is composed by a couple of notes **with consecutive number**, and containing at least L notes and at most R notes. Let us define the *pleasant value* of a super chord to be the sum of all pleasant value from each note in that chord. Two super chords are the same if and only if they contains same set of notes.

Little Zhik decides to create a song with k **different** super chords, and the *pleasant value* of a song is simply the sum of all pleasant value to all k super chords. Little Zhik is wondering any song with largest pleasant value he can create.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

First line contains four integers n, k, L, R, where n denotes the total number of notes ($1 \leq n \leq 100000$), k is the length of the song ($1 \leq k \leq 500000$), both L and R satisfies $1 \leq L \leq R \leq n$. From the second line to the $n+1$-th line, there are n numbers $A_1, \ldots A_n$ ($|A_i| \leq 1000$) indicating the pleasant values of note number $1, 2, \ldots n$ respectively. The test data is guaranteed that there is always at least a song containing k different super chords.

Output

For each test case please output an integer indicating the largest pleasant value that Little Zhik can create.

Sample Input

```
1
4 3 2 3
3
2
-6
8
```

Sample Output

```
11
```

(This page is almost empty.)

C. Conspiracy

Hostile Bitotia launched a sneak attack on Byteotia and occupied a significant part of its territory. The King of Byteotia, Byteasar, intends to organise resistance movement in the occupied area. Byteasar naturally started with selecting the people who will form the skeleton of the movement. They are to be partitioned into two groups: *the conspirators* who will operate directly in the occupied territory, and *the support group* that will operate inside free Byteotia.

There is however one issue — the partition has to satisfy the following conditions:

- Every pair of people from the support group have to know each other — this will make the whole group cooperative and efficient.
- The conspirators must not know each other.
- None of the groups may be empty, i.e., there has to be at least one conspirator and at least one person in the support group.

Byteasar wonders how many ways there are of partitioning selected people into the two groups. And most of all, whether such partition is possible at all. As he has absolutely no idea how to approach this problem, he asks you for help.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

The first line of the standard input holds one integer n ($2 \leq n \leq 5000$), denoting the number of people engaged in forming the resistance movement. These people are numbered from 1 to n (for the sake of conspiracy!). The n lines that follow describe who knows who in the group. The i-th of these lines describes the acquaintances of the person i with a sequence of integers separated by single spaces. The first of those numbers, k_i ($0 \leq k_i \leq n-1$), denotes the number of acquaintances of the person i. Next in the line there are k_i integers $a_{i,1}, a_{i,2}, \ldots, a_{i,k_i}$ — the numbers of i's acquaintances. The numbers $a_{i,j}$ are given in increasing order and satisfy $1 \leq a_{i,j} \leq n$, $a_{i,j} \neq i$. You may assume that if x occurs in the sequence a_i(i.e., among i's acquaintances), then also i occurs in the sequence a_x (i.e., among x's acquaintances).

Output

In the first and only line of the standard output your program should print out one integer: the number of ways to partition selected people into the conspirators and the support group. If there is no partition satisfying aforementioned conditions, then 0 is obviously the right answer.

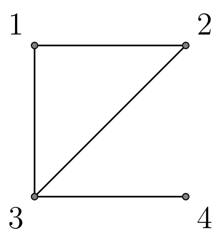
Sample Input

```
1
4
2 2 3
2 1 3
3 1 2 4
1 3
```

Sample Output

```
3
```

Explanation



There are three ways of partitioning these people into the groups. The group of conspirators can be formed by either those numbered 1 and 4, those numbered 2 and 4, or the one numbered 4 alone.

D. Dividing by Squares

A positive integer k is called square-divisible, if there is an integer $d > 1$ such that d^2 divides k. Now, giving an integer n, $1 \leq n \leq 10^{10}$, you are going to find the n-th smallest square-divisible integer.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case, there are an integer n in a line satisfying $1 \leq n \leq 10^{10}$.

Output

For each test case please output a line containing the n-th smallest square-divisible integer.

Sample Input

```
1
10
```

Sample Output

```
27
```

(This page is almost empty.)

E. Extra Driving Directions

Contrary to the popular belief, alien flying saucers cannot fly arbitrarily around our planet Earth. Their touch down and take off maneuvers are extremely energy consuming, so they carefully plan their mission to Earth to touch down in one particular place, then hover above the ground carrying out their mission, then take off. It was all so easy when human civilization was in its infancy, since flying saucers can hover above all the trees and building, and their shortest path from one mission point to the other was usually a simple straight line — the most efficient way to travel. However, modern cities have so tall skyscrapers that flying saucers cannot hover above them and the task of navigating modern city became quite a complex one. You were hired by an alien spy to write a piece of software that will ultimately give flying saucers driving directions throughout the city. As your first assignment (to prove your worth to your alien masters) you should write a program that computes the shortest distance for a flying saucer from one point to another. This program will be used by aliens as an aid in planning of mission energy requirements.

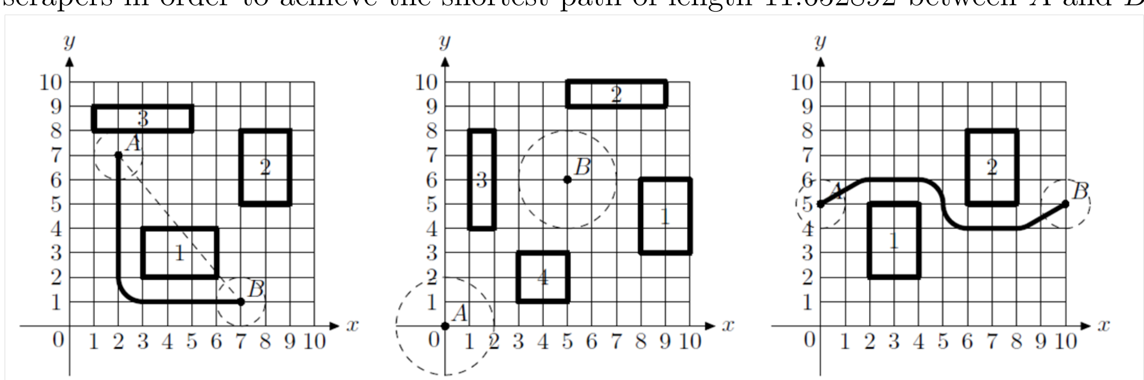
The problem is simplified by several facts. First of all, since flying saucer can hover above most of the buildings, you are only concerned with locations of skyscrapers. Second, the problem is actually twodimensional — you can look at everything "from above" and pretend that all objects are situated on OXY Cartesian plane. Flying saucer is represented by a circle of radius r, and since modern cities with skyscrapers tend to be regular, every skyscraper is represented with a rectangle whose sides are parallel to OX and OY axes.

By definition, the location of flying saucer is the location of its center, and the length of the path it travels is the length of the path its center travels. During its mission flying saucer can touch skyscrapers but it cannot intersect them.

At the first picture a flying saucer of $r = 1$ has to get from point A to point B. The straight dashed line would have been the shortest path if not for skyscraper 1. The shortest way to avoid skyscraper 1 is going around its top right corner, but skyscraper 2 is too close to fly there. Thus, the answer is to go around the bottom left corner of skyscraper 1 for a total path length of 10.570796.

In the second picture it is impossible for a flying saucer of $r = 2$ to get from point A to point B, since all skyscrapers are too close to fly in between them.

In the third picture flying saucer of $r = 1$ has to fly in a slalom-like way around two skyscrapers in order to achieve the shortest path of length 11.652892 between A and B.



Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

The first line of the input file contains integer numbers r and n ($1 \leq r \leq 100, 0 \leq n \leq 30$), where r is the radius of the flying saucer, and n is the number of skyscrapers. The next line contains four integer numbers x_A, y_A, x_B, and y_B ($-1000 \leq x_A, y_A, x_B, y_B \leq 1000$), where (x_A, y_A) are the coordinates of the starting point of the flying saucer's mission and (x_B, y_B) are the coordinates of its finishing point.

The following n lines describe skyscrapers. Each skyscraper is represented by four integer numbers x_1, y_1, x_2, and y_2 ($-1000 \leq x_1, y_1, x_2, y_2 \leq 1000, x_1 < x_2, y_1 < y_2$) — coordinates of the corners of the corresponding rectangle.

Skyscrapers neither intersect nor touch each other. Starting and finishing points of the flying saucer's mission are valid locations for flying saucer, that is, it does not intersect any skyscraper in those points, but may touch some of them.

Output

Write to the output file text "`no solution`" (without quotes) if the flying saucer cannot reach its finishing point from the starting one. Otherwise, write to the output file a single number — the shortest distance that the flying saucer needs to travel to get from the starting point to the finishing point. Answer has to be precise to at least 6 digits after the decimal point.

Sample Input

```
3
1 3
2 7 7 1
3 2 6 4
7 5 9 8
1 8 5 9
2 4
0 0 5 6
8 3 10 6
5 9 9 10
1 4 2 8
3 1 5 3
1 2
0 5 10 5
2 2 4 5
6 5 8 8
```

Sample Output

```
10.570796
no solution
11.652892
```

F. Fitting Pattern Painting

Lukas has a large grid. Initially, all squares of the grid are of white colour. Lukas has three patterns (numbered from one to three, starting with the left one):

```
XXXX        X.X.        X.X.
....        X.X.        .X.X
XXXX        X.X.        X.X.
....        X.X.        .X.X
```

He applies patterns on the grid. He chooses a rectangle, chooses a pattern and paints some squares with black color according to the pattern. He repeats this N times. When painted areas overlap, he obeys the OR rule. For example, if he chooses pattern 1 and then pattern 3 on a square 4×4, he gets

```
XXXX
.X.X
XXXX
.X.X
```

Your task is to compute number of black squares after Lukas has painted all rectangles. The pattern begins in the top left corner of a rectangle (the lowest x and the highest y coordinate).

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

In the first line of input, there is number N ($0 \le N \le 100000$). N lines follow. In each of these lines, there are five integers x_1, y_1, x_2, y_2, p, where (x_1, y_1) and (x_2, y_2) are the coordinates of grid points in two opposite corners of the rectangle, in which a pattern is applied, and p ($1 \le p \le 3$) is the number of the used pattern. Coordinates of rectangles do not exceed 10^9 in their absolute value and each rectangle is at least one unit high and wide.

Output

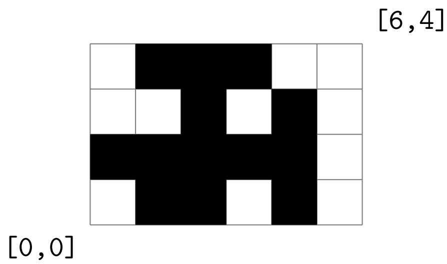
For each test case the only line of output should contain number of black squares.

Sample Input

```
1
3
0 0 3 2 3
1 4 4 1 1
2 3 6 0 2
```

Sample Output

```
13
```



G. Gathering Plots

We call any sequence of points in the plane a *plot*. We intend to replace a given plot $(P_1, \ldots, P_n)$ with another that will have at most m points ($m \leq n$) in such a way that it "resembles" the original plot best.

The new plot is created as follows. The sequence of points $P_1, P_2, \ldots, P_n$ can be partitioned into s ($s \leq m$) contiguous subsequences:

$$(P_{k_0+1}, \ldots, P_{k_1}), (P_{k_1+1}, \ldots, P_{k_2}), \ldots, (P_{k_{s-1}+1}, \ldots, P_{k_s}),$$

where $0 = k_0 < k_1 < k_2 < \ldots < k_s = n$, and afterwards each subsequence $(P_{k_{i-1}+1}, \ldots, P_{k_i})$, for $i = 1, \ldots, s$ is replaced by a new point Q_i. In that case, we say that each of the points $P_{k_{i-1}+1}, \ldots, P_{k_i}$ has been *contracted* to the point Q_i. As a result a new plot, represented by the points $Q_1, \ldots, Q_s$, is created. The measure of such plot's resemblance to the original is the maximum distance of all the points $P_1, \ldots, P_n$ to the point it has been contracted to:

$$\max_{i=1,\ldots,s} \left(\max_{j=k_{i-1}+1,\ldots,k_i} (d(P_j, Q_i)) \right),$$

where $d(P_j, Q_i)$ denotes the distance between P_j and Q_i, given by the well-known formula:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

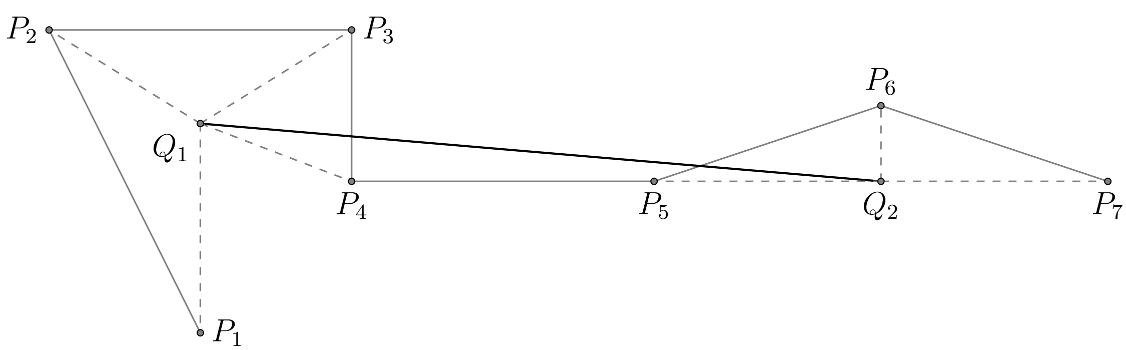


Figure 1: An exemplary plot $(P_1, \ldots, P_7)$ and the new plot (Q_1, Q_2) where $(P_1, \ldots, P_4)$ are contracted to Q_1 whereas (P_5, P_6, P_7) to Q_2.

For a given plot consisting of n points, you are to find the plot that resembles it most while having at most m points, where the partitioning into contiguous subsequences is arbitrary. Due to limited precision of floating point operations, a result is deemed correct if its resemblance to the given plot is larger than the optimum resemblance by at most 0.000001.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

In the first line of the standard input there are two integers n and m, $1 \leq m \leq n \leq 100000$, separated by a single space. Each of the following n lines holds two integers, separated by a single space. The $(i+1)$-th line gives x_i, y_i, $-10^6 \leq x_i, y_i \leq 10^6$, denoting the coordinates (x_i, y_i) of the point P_i.

Output

For each test case, in the first line of the standard output one real number d should be printed out, the resemblance measure of the plot found to the original one. In the second line of the standard output there should be another integer s, $1 \leq s \leq m$. Next, the following s lines should specify the coordinates of the points $Q_1, \ldots, Q_s$, one point per line. Thus the $(i+2)$-th line should give two real numbers u_i and v_i, separated by a single space, that denote the coordinates (u_i, v_i) of the point Q_i. All the real numbers should be printed with at most 15 digits after the decimal point.

Sample Input

```
1
7 2
2 0
0 4
4 4
4 2
8 2
11 3
14 2
```

Sample Output

```
3.00000000
2
2.00000000 1.76393202
11.00000000 1.99998199
```

H. Honey Sweets

To celebrate Children's Day a mother of three brothers — Anton, Dmytro and Borys — gave them loads of sweets. The treats are packed in n boxes, there are a_i sweets in the i-th box. The brothers want to share the boxes justly. They agreed to the following rules:

- an older brother cannot receive more sweets in total than a younger one (Anton is older than Dmytro, and Dmytro is older than Borys),
- the difference between the total number of sweets given to Anton and the total number of sweets given to Borys should be minimal.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

In the first line of the standard input there is an integer n ($3 \le n \le 24$), denoting the number of boxes. The second line consists of n positive integers a_i ($1 \le a_i \le 10^9$), denoting the number of sweets in the boxes.

Output

In the only line of each output you should write one integer — the difference between the numbers of sweets given to Anton and Borys.

Sample Input

```
2
4
5 4 7 6
4
4 5 7 6
```

Sample Output

```
3
3
```

(This page is almost empty.)

I. Statisticians

Statisticians like to create a lot of statistics. One simple measure is the mean value: the sum of all values divided by the number of values. Another is the median: the middle among all values when they have been sorted. If there are an even number of values, the mean of the two middle values will form the median.

These kinds of measures can be used for example to describe the population in a country or even some parts of the population in the country. Anne Jensen, Maria Virtanen, Jan Hansen, Erik Johansson and Jn rsson want to find a statistical measurement of how many statisticians there are in the Nordic countries. To be more precise, they want to

nd out how many statisticians there are per unit area. As the population in the Nordic countries are well spread out they will try the new measurement MAD, Median of All Densities. First put a square grid on the map. Then draw a rectangle aligned with the grid and calculate the density of statisticians in that area, i.e. the mean number of statisticians per area unit. After that, repeat the procedure until all possible rectangles have been covered. Finally the MAD is the median of all statistician densities.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

The first line of the input contains of two space separated numbers h and w describing the height and width of the square grid, where $1 \leq h \leq 140$ and $1 \leq w \leq 120$. The next line contains two space separated numbers a and b which are the lower and upper bound of the allowed rectangle areas, i.e. $1 \leq a \leq rectanglearea \leq b \leq w \times h$. Then there will follow h lines with w space separated numbers s describing the number of statisticians in each square of the map, $0 \leq s \leq 10000$. There will always exist a rectangle which area is in $[a, b]$.

Output

The output contains of one line with the MAD. The number should be printed in number of statisticians per square and contain three decimals. The absolute error must be $< 10^{-3}$.

Sample Input

```
2
2 3
2 4
6 1 4
2 7 1
4 2
1 8
6 5
2 5
2 9
7 13
```

Sample Output

```
3.667
5.250
```

J. Jewellery

Some jewels are placed in a rectangular grid on an infinite plane. If the j-th character of the i-th row is 'R', 'G' or 'B', there is a red, green or blue jewel at the center of the j-th cell of the i-th row of the grid, respectively. Jewels of the same color cannot be distinguished.

Alice draws two squares on this plane. Their sides must be parallel to the axis and have length k. She is allowed to draw squares so that some part of them are on the outside of the grid. Then she will get the jewels which lie in the *inside* of at least one of the two squares. Note that the *inside* of a square does not contain its boundary. She wants to make a chain of jewels, using some of the jewels she will get. She does not have to use all of the jewels she will get. A chain is a row of one or more jewels. Chains are non-directional. For example, chains R-G-B and B-G-R are considered equal. How many different chains are possible, considering all the chains from all possible square locations? Output the answer modulo 1000000009.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

First line contains three integers m, n, k ($1 \leq m, n, k \leq 44$) indicating the size of the grid. Next m lines contains strings of length n denoting the rectangular grid. Every character on the grid will be one of 'R', 'G', 'B'.

Output

For each test case please output the answer in a line.

Sample Input

```
6
1 3 1
RGB
1 3 2
RGB
1 3 10
RGB
4 5 2
RRRRR
RGGRR
RGGGG
RRRGG
10 10 6
RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
```

```
RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
RRRRRRRRRR
5 8 4
RRRGGGGG
RRRGGGGG
RRRGGBBB
GGGGGBBB
GGGGGBBB
```

Sample Output

```
6
9
9
280
68
613435159
```

Königsberg's Bridges

This problem is somewhat unrelated to Königsberg, it's about somewhere called "San Bytecisco" in Polandemerica.

San Bytecisco is a beautifully situated coastal town. It consists of n small, yet densely populated islands, numbered from 1 to n. Certain islands are connected with bridges, used for (bidirectional) road traffic. Each pair of islands can be connected with at most one bridge. The islands are connected in such a way that every island can be reached from every other by using the bridges only.

Byteasar and Bytie are going for a bike trip in San Bytecisco. The will start their ride at the island no. 1. They intend to visit every island, while passing along every bridge once and ending the trip where it began, i.e., the island no. 1. Being quite seasoned riders, they expect some serious trouble from... the wind! After all, it is very windy along the coast, and especially so on the bridges between the islands. Obviously, depending on its speed and direction, the wind makes it hard to cross the bridge *in different extent for either direction*. For simplicity we will assume for every bridge and direction of crossing, the opposing wind speed is constant.

Help Byteasar and Bytie to find a route as they desire that will in addition be the least tiresome. Byteasar and Bytie agreed on the maximum opposing wind speed as a measure of a route's tiresomeness.

Input

The first line contains an integer T indicating the number of test cases in the input file. For each test case:

In the first line of the standard input there are two integers separated by a single space: n and m ($2 \leq n \leq 1000, 1 \leq m \leq 2000$), denoting the number of islands and the number of bridges in San Bytecisco respectively. The islands are numbered from 1 to n, while the bridges from 1 to m. The following lines specify the bridges. The line no.$(i+1)$ contains four integers a_i, b_i, l_i, p_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq l_i, p_i \leq 1000$), separated by single spaces. These denote that the bridge no.i connects the islands no.a_i and b_i. The opposing wind speeds are l_i when one goes moves from a_i to b_i, and p_i if one goes from b_i to a_i.

Output

For each test case:

If there is no route satisfying the requirements of the daring two riders, the first and only line of the standard output should hold the word "`NIE`" (*no* in Polish). Otherwise, the output should hold the maximum opposing wind speed for that route, i.e., the number we wish to minimize.

Sample Input

```
1
4 4
1 2 2 4
2 3 3 4
3 4 4 4
4 1 5 4
```

Sample Output

```
4
```