Skyly's Contest I

2012-04-29 @ NTU Online Judge

Table 1: Problem Set Information

	Problem Title	**Runtime Limits**
Problem A	Ancient Mysterious Puzzle	5 seconds
Problem B	Backup Server	10 seconds
Problem C	Classic Collection	10 seconds
Problem D	Defense with Towers	5 seconds
Problem E	Equal to Zero	5 seconds
Problem F	Formula Racing	10 seconds
Problem G	Grouping to Race	5 seconds
Problem H	Hyper-broadcast Satellite	5 seconds
Problem I	Income Inequality	5 seconds
Problem J	Just Teach Me	5 seconds

*Special Thanks to **csferng** for the great help to this contest*

(This page is intentionally left blank.)

Problem A
Ancient Magical Puzzle

time limit: 5 seconds

Taking an exciting adventure is a wonderful dream of almost every little boys, and so is Kelvin. Kelvin is a smart boy who wants to travel around the world, and he wants to see the "World Wonders" as many as possible, after he played a classic strategic game *AOE2*. Among all the Wonders of ancient civilizations, little Kelvin especially loves far-east Wonders, and he wishes that he can have an exciting adventure within the abstruse and mysterious atmosphere.

"Time flies like an arrow," a past Chinese sage said. Our little boy Kelvin is a well-known, professional treasure hunter now. Although his skills are getting mature day by day, and he is not a little boy that he used to be anymore, he still has the dream about travelling the mysterious Wonders at far-east.

One day, Kelvin found a magical old book with a special star-shaped front cover. With his marvelous intuition, he noticed that it is the book about one of the far-east Wonders, the "Concrete Temple," which is built by Kakalian at about 514 B.C. After some messy works on the text-translation, Kelvin found that the old book indicated the location of the "Concrete Temple." However, in order to get in the temple, all the "guests" need to solve a puzzle to enable the wooden elevator. Otherwise, traps will be enabled, and all the people will stuck in the underground forever.

According to the statement of the old book, the puzzle is made by magical stones, which glitter with many colors. The puzzle is formed by a square stone-made board with grids within n rows and n columns, say an n by n boards. For every single grid, there is a magical stone within it. To overcome the challenge, Kelvin need to solve the puzzle by doing some operations — to let all the magical stones in the given pattern(some pre-selected positions) be the same color. The types of operations that Kelvin can do are stated as following:

(1) Press the i^{th} mysterious blue button to reverse the i^{th} row of magical stones.

(2) Press the j^{th} mysterious green button to reverse the j^{th} column of magical stones.

(3) Press the mysterious red button to flip the whole state on the board with respect to the main diagonal.

(4) Press the mysterious yellow button to flip the whole state on the board with respect to the secondary diagonal(the counterdiagonal).

Based on the rule referenced by the old book, Kelvin needs to perform "minimum" times of operations (each type of operation costs 1 step, of course, he can perform none of them if the initial state is already fit the required pattern) to open the gate of the wooden elevator. Since Kelvin is extremely smart, so he can definitely solve the given puzzle if the puzzle is solvable. But what if the given puzzle is not able to solve in anyway? If that is just completely a trap for its guests? The experience of adventuring into a lot of dangerous places reminds Kelvin that there has possibility of it.

However, the old book did not mention about the size of that stone-made board. That is, n is unknown for Kelvin now. Please write a program to help Kelvin identifying whether the given puzzle is solvable or not.

■ Input Format

The first line of input contains an positive integer $T(\leq 50)$, which denotes the number of puzzles that should be identified.

The following lines describe the puzzles one by one. For every puzzle, the first line has an integer n ($2 \leq n \leq 20$), then have two $n \times n$ matrices: the initial state of the board, and the required pattern. For the first matrix, there is an integer c_{ij} ($1 \leq c_{ij} \leq 9$) which stands for the color of magical stone for every entry in the initial state of the board. Two magical stones that have the same color will correspond to the same integer.

For the second matrix, the positions that are pre-selected are marked by `#`, and the other positions are marked by `*`. That is, all the positions marked by `#` need to be cover by the magical stones with the same color. The old book does not guarantee the number of selected positions.

■ Output Format

For each given puzzle, please output the result that if Kelvin can solve it.

If the given puzzle is solvable by Kelvin, then print a single line "`YES`" (without the quotes). Otherwise, please print a single line"`NO`" (without the quotes).

■ Sample Input

```
3
5
1 2 3 4 5
5 4 3 2 1
1 2 3 4 5
5 4 3 2 1
1 2 3 4 5
# * * * *
# * * * *
# * * * *
# * * * *
# * * * *
3
1 2 2
1 1 2
1 2 1
# # *
* # *
# # #
4
7 3 2 2
6 8 9 3
7 9 8 3
4 4 5 1
* * * *
# * * *
* * * *
* # # *
```

■ Sample Output

```
YES
NO
YES
```

■ Sample Explanations

For the first example, smart Kelvin will press the second blue button once and the fourth blue button once to solve the puzzle.

For the second example, Kelvin cannot fit the pattern anyway because neither color 1 nor 2 have six same-color magical stones.

For the last example, smart Kelvin will simply press the red button once to solve the puzzle.

Problem B

Backup Server

time limit: 10 seconds

Small-Small Kuo is working as the system administrator in the workstation lab in NTU-CSIE. He is responsible for backing up images of different servers and rolling back to a previous version if needed.

Currently Kuo is struggling with the problem of lacking of space to store the images for recovery. He decides to transfer the images to a new cluster, which is more powerful and reliable. Unfortunately, if something happens during the transfer, the images may be lost and cannot be recovered! Therefore, the procedure of transfer must be carefully planned.

Kuo has N back-up spaces, where the i-th back-up space stores the image of server a_i. He has decided to split the transfer into stages. At each stage he would like to make sure that for some l and r, the back-up spaces l, $l+1$, $\cdots$, r are not affected by the transfer and can be used for rolling back k different servers once a problem occurs.

In order to plan the optimal transfer procedure, Kuo needs your help to answer this question: For a given l, what is the minimum r such that the back-up spaces l, $l+1$, $\cdots$, r are able to roll back k different servers.

■ Input Format

The first line of the input file contains an integer T, the number of test cases. Two consecutive test cases are separated by a blank line.

The first line of each test case contains two integers N and M, indicating the number of back-up spaces and the number of servers ($1 \le N, M \le 10^5$). The second line consists of N numbers, a_1, $\cdots$, a_N, where a_i denoting the server which image stored in the i-th back-up space ($1 \le a_i \le M$).

The third line contains an integer Q, the number of queries ($1 \le Q \le 10^5$). To process the queries, it is necessary to maintain a number p, which is 0 initially. Each query is given on one line with a pair of integers x and y ($1 \le x \le N$, $1 \le y \le M$). The query is as follows: $l = ((x+p) \mod N) + 1$, $k = ((y+p) \mod M) + 1$. Suppose the answer of this query is r, you should assign the value of r to p for the next query.

■ Output Format

For each query please output an integer: the required minimum r, or 0 if no such r exists. Please output a blank line after each test case.

■ Sample Input

```
7 3
1 2 1 3 1 2 1
4
7 3
7 1
7 1
2 2
```

■ Sample Output

```
1
4
0
6
```

Problem C
Classic Collection

time limit: 10 seconds

There are N fantastic wooden carving, the shopkeeper wants to sale a "Classic Collection" to the real carving-lovers. Each carving can be considered as a polygon which has edges paralleled to the x-axis or the y-axis, and length of each edge is a positive integer. Moreover, each wooden carving has a "beauty factor", b_i, and you want to choose some of them into the "Classic Collection" (choose a subset of carvings).

In order to package the wooden carvings that you chose to sell, you need a square box to pack them up. However, a larger box costs more. More precisely, a box with side-length x cost cx, where c is a costing multiplier. You need to put all of the carvings you chose into the box with an non-overlapping arrangement. You can rotate 90, 180, or 270 degrees toward a carving, and you can also flip a carving.

Now, we define the "classic value" of every possible "Classic Collection" as the sum of "beauty factor" of the carvings that you chose minus the cost of the outer box. The shopkeeper is interesting about the maximum "classic value" that can be achieved. Notice that you should choose at least one wooden carving for the"Classic Collection."

■ Input Format

The first line of input contains an integer $T(T \leq 20)$, which denotes that the input consists of T test cases.

For each test case, the first line has two integers, N and c, denoting that there are $N(2 \leq N \leq 10)$ wooden carvings in total, and the costing multiplier $c(1 \leq c \leq 1000)$. The second line contains N integers, the i-th integer denotes the "beauty factor" b_i. The following N lines describe the shape of N wooden carvings. For the i-th line, we describe the shape of the i-th wooden carving by starting drawing from the origin (i.e., $(x = 0, y = 0)$), then the following integers with the odd indices(1-based) denote the x-differences, and the following integers with the even indices denote the y-differences. For example, "`+1 +1 +1 +1 -2 -2`" denote the polygon containing edges $\{(0,0),(1,0)\}$, $\{(1,0),(1,1)\}$, $\{(1,1),(2,1)\}$, $\{(2,1),(2,2)\}$, $\{(2,2),(0,2)\}$, and $\{(0,2),(0,0)\}$.

You can assume that every carving is a simple closed polygon, and a box with side-length equals to 10^5 can package all the given wooden carvings.

■ Output Format

For each test case, print out a single line that contains an integer which indicates the maximum possible "classic value."

■ Sample Input

```
3
3 2
10 5 50
+1 +3 -1 -3
+50 +1 -50 -1
+2 +2 -2 -2
2 5
3 4
+1 +1 +1 +1 -2 -2
+1 +1 +1 +1 -2 -2
2 1
3 4
+1 +1 +1 +1 -2 -2
+1 +1 +1 +1 -2 -2
```

■ Sample Output

```
54
-6
4
```

Problem D
Defense with Towers

time limit: 5 seconds

The Bright Kingdom is under attack now! Estelle, the beautiful princess of the Bright Kingdom, is now asking for your help!

There was very peaceful in the Bright Kingdom during the past several years. The people lived in there were just like lived in the Arcadia. However, since a month ago, lots of monsters appeared just like all of sudden, and "they" started to attack the Bright Kingdom immediately! After some crucial battles, the Bright Kingdom already lost many soldiers and spent a lot of money and food. It seemed to be hopeless to conquer the enemy, since there are so many monsters that we did not know where they came from. Unfortunately, Estelle's father, the king of Bright Kingdom, had been killed by a half-orc in a massive encounter, and that is why you are the only hope for Estelle now.

During the war, the military of Bright Kingdom has built N towers to defend the attacks of the enemy. Each tower has an attacking range, and it can eliminate the monsters that are "totally covered" by its range. According to the information from the scouts of the kingdom, a huge wave of monsters is coming, and it's time to use these towers!

To make the problem simpler, we can consider our battle field as a plane with Cartesian coordinates. The main gate of the Bright Kingdom is at $y = 0$, and the avenue in front of the gate has width $2d(d > 0)$, from $x = -d$ to $x = d$. The top-end of the battle field is considered to be line $y = y_{max}$. That is, the field we cared is a rectangular area bounded by $y = 0$, $y = y_{max}$, $x = -d$, and $x = d$.

The monsters can be considered as some rectangular objects. There are M monsters in total. The j-th monster has the length l_j and the same width w, and it will be appeared in the top-end of the battle field where centralized at x-coordinate $x = x_j$ (that is, $x_{Left} = x_j - w/2$, $x_{Right} = x_j + w/2$) at the a_j-th second from the beginning, then it will move down straightly with the fixed speed $v = 1$ unit per second. Notice that there might be several monsters overlapped at some times.

But the things are not quite easy, since the energy that Bright Kingdom has is not such enough now. However, we need energy to trigger the towers. More precisely, the i-th tower is placed at $y = y_i$ and has a circle range with radius r_i. It need e_i unit(s) of energy to enable it for the next t_i seconds. You can "turn on" the tower(s) many times. Moreover, for our towers, some of them are placed on the left side ($x = -d$) of the avenue; the others are placed on the right side($x = d$) of the avenue. In order to make the Bright Kingdom survived, you need to decide the strategy about how to use these R towers to defeat all the monsters that will be arrived soon.

Notice that you cannot union the ranges of two distinct towers to eliminate a monster. A monster can be slain if and only if it is totally covered by the range of a single tower. (Of course, if there are two or more towers can "totally" cover a monster, then it will be slain, too.)

■ Input Format

The input consists of multiple test cases, the first line of input contains an integer $T (T \leq 50)$ which denotes the number of test cases.

For each test case, the first line has five integers, N, M, d, y_{max}, w, separated by the spaces. For the following N lines, the i-th line contains four integers y_i, r_i, e_i, t_i and a character s_i (the tower is placed at the left if $s_i =$'`L`', and at the right if $s_i =$'`R`') for the i-th tower. For the last M lines, the j-th line have three integers x_i, l_i, a_i of the j-th monster. ($1 \leq N, M \leq 15$, $1 \leq d \leq 100$, $1 \leq y_{max} \leq 100$, $1 \leq w \leq 2d$, $0 \leq y_i \leq y_{max}$, $1 \leq r_i \leq 2d$, $1 \leq e_i, t_i \leq 10^6$, $1 \leq l_i \leq y_{max}$, $-d \leq x_i - w/2$, $x_i + w/2 \leq d$, $0 \leq a_i \leq 10^6$)

■ Output Format

For each test case, please output the minimum usage of energy to defeat all the monster(s). If cannot achieve the goal, please print a single line, "Run!! Estelle!!" (without the quotes).

■ Sample Input

```
3
2 2 2 5 1
3 2 10 10 L
2 2 15 10 R
-1 1 7
1 1 19
1 3 3 8 2
4 4 7 20 R
2 1 0
2 2 1
2 2 25
1 1 100 8 1
5 150 12 12 L
195 1 999997
```

■ Sample Output

```
25
14
Run!! Estelle!!
```

(This page is intentionally left blank.)

Problem E
Equal to Zero

time limit: 5 seconds

You likely have seen that $x(\sin^2 x + \cos^2 x) - x = 0$, and you may have seen that $\sin(2x) - 2\sin x \cos x = 0$. But did you know that $\tan(2x)(x - x\tan^2 x) - 2x\tan x = 0$? Would you believe that $\sin(2x) - 2\cos x = 0$? That last one is false, but don't just take our word for it; you should write a program that determines whether an algebraic expression simplifies to zero (whenever it is defined).

■ Input Format

The input consists of multiple test cases, each on one line. Each test case is a math expression which consists of variable `x`, operators `+`, `-`, `*`, and functions `sin(·)`, `cos(·)`, `tan(·)`.

You may assume that the input is valid. The length of each line will be at most 300 characters. Function arguments can contain functions, so `sin(sin(x))` is valid, but the recursion will not go any deeper than this.

■ Output Format

For each test case, print out a single line that contains "`Identity`" if the expression is always zero, and "`Not an identity`" otherwise (quotes added for clarity).

■ Sample Input

```
(sin(x)*sin(x)+cos(x)*cos(x))*x-x
sin(x)*cos(x)+sin(x)*cos(x)-sin(x+x)
tan(x+x)*(x-tan(x)*tan(x)*x)-tan(x)*x-tan(x)*x
sin(x+x)-cos(x)-cos(x)
```

■ Sample Output

```
Identity
Identity
Identity
Not an identity
```

(This page is intentionally left blank.)

Problem F
Formula Racing

time limit: 10 seconds

Formula racing is one of the most exciting sport in the world. The term "formula racing" refers to various forms of open wheeled single seater motorsport. The racing series are called "formula" because of the "rules" constrained on the specification of racing car. (So its name has nothing related with mathematical formulas :-)) There are many series of formula racing that partition into many tiers.

Among so many different series, the "Formula One" (as known as the "F1") is probably the most well-known series. A Formula One race or a Grand Prix is a sporting event which takes place over three days (usually Friday to Sunday), with a series of practice and qualifying sessions prior to a race on Sunday, though the structure of the weekend has changed numerous times over the history of the sport. All of the drivers and constructors compete for the winner for every Grand Prix, and make lots of efforts to obtain "points" for the point system. The driver/constructor that has the highest point will become the World Champion of that year.

Now, a formula racing series called "Formula I" is hold. which is similar to the Formula 1. In the Formula I series, there has only one tyre supplier, and the rule is that the tyres you can use for the whole year are all given in the beginning of the year. That is, you cannot get more new tyres during the season. The tyre supplier provides four types of tyre: Super-Soft, Soft, Medium, and Hard. Each type of tyre has different speed and durability on the tracks. (In this problem we do not consider the Intermediate and Wet tyres.)

A new racing season is coming! There are $S (S \leq 20)$ Grand Prix championship in total this year, and you need to complete d_i kilometers to complete the i-th site. For every site, you need to decide how many tyres for each class to use. However, this is important because you want to win in every sites, but you also want to retain the "good tyres" for the remaining sites. Fortunately(?), the great (and cute?) racing engineer Kakaen Huang has wonderfully estimated the racing time for the possibly "second-placed" driver. That is, you have already know that if you can complete the race within time t_i, then you will definitely win! (Since Kakaen is so smart, he never makes mistakes)

You are the racing strategist on the team, and are given the information that the tyre-types we can use in each Grand Prix. Also, you are given the speed and durability of each types for each Grand Prix. Please compute that how many tyres we need to use at least, for this year, in order to be the winner in all the sites.

Notice that you can use some "partial pieces" of tyres. That is, you can use a new tyre for a Grand Prix and then use it "again" in the next Grand Prix with the reduced durability, if the tyre still can use. For example, you can use "0.7 pieces of" Soft tyre in a Grand Prix and then use the remaining "0.3 pieces of" Soft tyre in the next Grand Prix event. Also, you can assume that the pit-time (the time spent on car-maintenance) is too short so that it can be neglected.

■ Input Format

The first line of input contains an integer $T (T \leq 25)$, which denotes that the input consists of T test cases, which are stated as following.

For each test case, which describes a scenario above, the first line has five integers, S, and the numbers of tyres that are given, for Super-Soft, Soft, Medium, and Hard, respectively.

Then, there are S lines for each Grand Prix this year. The i-th line contains ten floating numbers: the total distance for this Grand Prix(in meters), the time limit for winner(in minutes), and the speed(in meters per second) of each type of tyres on this track, for Super-Soft, Soft, Medium, and Hard, respectively; the last four floating numbers denote the durability(in meters) of each type of tyres on this track, for Super-Soft, Soft, Medium, and Hard, respectively. (If the durability is 5000 meters, then it costs 10% of one tyre for 500 meters.)

If the speed of a type of tyres on some tracks is negative, then it means that you cannot use that type for that Grand Prix championship.

■ Output Format

For each test case (scenario), please output a floating-point number indicated that how many tyres we need to use at least, in order to win all of the races.

■ Sample Input

```
(Please refer to the problem at NTUJ)
```

■ Sample Output

```
(Please refer to the problem at NTUJ)
```

Problem G
Grouping to Race

time limit: 5 seconds

You own $2N$ horses, and the i-th horse can run i kilometers in 10 minutes. Now you are going to held a tournament of N races. The horses are split into 2 groups, $\mathcal{A}$ and $\mathcal{B}$, each with N horses. A race is contested by two horses, one from each group. The speed of the horses determines the winner of a race. Of course, each horse can participated in only one race.

Your younger brother, who is always very kind, volunteered to help. Thus you divide the work into two parts: At first, you will decide which horse goes to which group. After that, your brother will tell you which horse in group $\mathcal{A}$ and which horse in group $\mathcal{B}$ should contest in the same race.

As you know, your brother loves 'A' so much and always get straight A's in school, so he will arrange the races such that group $\mathcal{A}$ will win as many races as possible. For example, if $\mathcal{A} = \{1, 3, 5\}$ and $\mathcal{B} = \{2, 4, 6\}$, he may arranges the races like this: 3-vs-2, 5-vs-4, 1-vs-6. In this case group $\mathcal{A}$ can win 2 races.

You, as a naughty boy, would like to tease your brother a bit. You decide to split the horses into groups in a way that group $\mathcal{A}$ will lose at least K races no matter how the races are arranged. How many ways of splitting can achieve this goal?

■ Input Format

The input consists of multiple test cases, each on one line. Each line contains two integers N and K, separated by a white space. $1 \leq N \leq 10^{16}$, $0 \leq K \leq N$.

■ Output Format

For each test case, please output the number of ways to split the horses as required, modulo 102217.

■ Sample Input

```
1 1
1 0
2 1
```

■ Sample Output

```
1
2
4
```

Problem H
Hyper-broadcast Satellite

time limit: 5 seconds

The Nanoha Broadcasting Corporation has invented a new type of ratio-broadcast satellite. Their researching lab developed a new formation of transmission, which was called "hyper-broadcast", and they believe that it will become the new generation of broadcast technology.

To take some advantages from this new broadcast technology. The Nanoha Broadcasting Corporation wants to launch a "hyper-broadcast" satellite, to show the amazing covering ability to the people. The Nanoha Broadcasting Corporation first want to service the islands, which usually have bad signal-qualities, on the ocean. In order to show the power of their new broadcast technology, they want to cover the total area of islands as much as possible. The area that a satellite can cover is a circle with radius equal to r.

However, since the actual orbit of satellite might have a little difference(error) from the position we want (the best location). Therefore, the center of the round broadcasting area is possibly anywhere within a rectangular range $x_1 \le x \le x_2$, $y_1 \le y \le y_2$. Assume that the probability of the center within the pre-calculated rectangular range follows a uniform distribution. That is, every point in that rectangular area has the same probability to become the center of the round broadcasting range.

To make the problem simpler, we can treat the given islands as some simple polygons. You can assume that the polygons are mutually non-intersecting.

Now the CEO of Nanoha Broadcasting Corporation wants to know the mathematical expectation value of the total covering area of the islands (i.e., the average total covering area), and you are an engineer in the Nanoha Broadcasting Corporation. Can you solve this challenging problem?

■ Input Format

The input consists of multiple test cases, the first line of input contains an integer $T (T \le 20)$ which denotes the number of test cases.

For every test case, the first line has one integers $N (1 \le N \le 10)$, and then five floating-point numbers $r (0 < r \le 10^6)$, x_1, y_1, x_2 and y_2. $(-10^6 \le x_1, y_1, x_2, y_2 \le 10^6)$ There are N following lines, for the i-th line, there first comes an integer $v_i (3 \le v_i \le 50$, for $1 \le i \le N)$ which indicates the number of vertices that the i-th polygon(island) contains; then there are v_i pairs of floating-point numbers x_{ij}, y_{ij} $(1 \leq j \leq v_i)$ that denote the coordinates of each vertex, given in either clockwise or counter-clockwise direction. All the input numbers are separated by the space(s).

■ Output Format

For each test case, please print a single line with a floating number that denotes the mathematical expectation value of the total covering area.

Answers with less than 10^{-2} absolute or relative error will be accepted.

■ Sample Input

```
2
2 1.0 0.1 0.25 0.1 0.25
4 0.1 0.25 1.1 0.25 1.1 1.25 0.1 1.25
4 0.1 0.25 -0.9 0.25 -0.9 -0.75 0.1 -0.75
1 5.0 -1.0 -1.0 1.0 1.0
3 -514.0 -514.0 0.0 514.0 514.0 0.0
```

■ Sample Output

```
1.570796
78.539816
```

Problem I

Income Inequality

time limit: 5 seconds

"Income inequality metrics"(or "income distribution metrics") are used by social scientists to measure the distribution of income, and economic inequality among the participants in a particular economy, such as that of a specific country or of the world in general. While different theories may try to explain how income inequality comes about, income inequality metrics simply provide a system of measurement used to determine the dispersion of incomes. Nowadays, Income distribution has always been a central concern of economic theory and economic policy.

The "Gini coefficient" (also known as the "Gini index" or "Gini ratio") is a measure of statistical dispersion developed by the Italian statistician and sociologist Corrado Gini. The Gini coefficient measures the inequality among values of a frequency distribution (for example, levels of income). A low Gini coefficient indicates a more equal distribution, with 0 corresponding to complete equality, while higher Gini coefficients indicate more unequal distribution, with 1 corresponding to complete inequality. To be validly computed, no negative goods can be distributed. Thus, if the Gini coefficient is being used to describe "household income inequality", then no household can have a negative income.

The Gini coefficient, G, is usually defined mathematically based on the "Lorenz curve", which plots the proportion of the total income of the population (y-axis) that is cumulatively earned by the bottom $x\%$ of the population (see the figure in the next page). The line at 45 degrees represents perfect equality of incomes. The Gini coefficient can then be thought of as the ratio of the area that lies between the line of equality and the Lorenz curve (marked 'A' in the diagram) over the total area under the line of equality (marked 'A' and 'B' in the diagram). (i.e., $G = A/(A+B)$)

To compute G, since $A + B = 0.5$, the Gini index, $G = A/(0.5) = 2A = 1 - 2B$. If the Lorenz curve is represented by the function $Y = L(X)$, the value of B can be found with integration, then we have

$$G = 1 - 2\int_0^1 L(X)dX.$$

However, the value of $L(X)$ is usually difficult to calculate. For a probability density function f(x) with the cumulative distribution function F(x), the Lorenz curve L(F(x)) is given by:

$$L(F(x)) = \frac{\int_{-\infty}^{x} tf(t)dt}{\int_{-\infty}^{\infty} tf(t)dt}$$

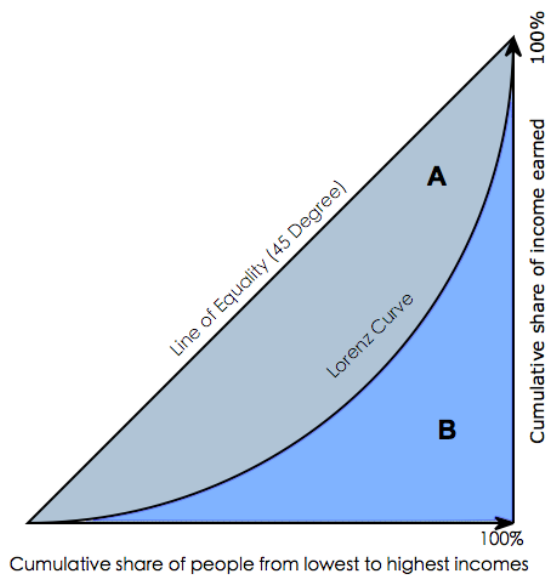

Figure I-1, Graphical representation of the Gini coefficient

Fortunately, the Gini coefficient can be calculated if you know the mean of a distribution, the number of people (or percentiles), and the income of each person (or percentile). Princeton development economist Angus Deaton simplified the Gini calculation to one easy formula:

$$G = \frac{N+1}{N-1} - \frac{2}{N(N-1)\mu}\left(\sum_{i=1}^{n} P_i X_i\right)$$

, where μ is mean income of the population, P_i is the income rank P of person i, with income X_i, such that the richest person receives a rank of 1 and the poorest a rank of N. If there are two people having the same amount of income, then the person with smaller index has the smaller rank.

Now, you are given the income of N randomly selected people in this society. In this problem, we treat $\mu = \bar{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i$ for an approximation. Please compute the approximated Gini coefficient by the Deaton formulation.

■ Input Format

The input consists of multiple test cases, the first line of input contains an integer $T(T \leq 100)$ which denotes the number of test cases.

Every test case occupies two lines in the input, the first line contains an integer $N(2 \leq N \leq 50000)$, and the second line has N integers which denote the income of each person, from the smaller index to the larger, respectively, separated by the space(s). All the income values are scaled within $[0, 100000]$.

■ Output Format

For each test case, please output the approximated Gini coefficient computed by the Deaton formulation.

Answers with a relative or absolute error at most 10^{-6} will be accepted.

■ Sample Input

```
7
10
1 2 3 4 5 6 7 8 9 10
9
9 8 1 2 3 5 10 2 7
7
514 513 512 511 510 509 508
2
1000 0
3
0 1 100000
4
64 64 128 128
2
255 255
```

■ Sample Output

```
0.333333
0.388298
0.002609
1.000000
0.999990
0.222222
0.000000
```

(This page is intentionally left blank.)

Problem J

Just Teach Me

time limit: 5 seconds

"Home-teaching" is a popular way to learn anything you want at home in Taiwan. For example, you can learn English, French, Mathematics, Computer Sciences, Economics, Physics, Chemistry, etc. You can just make a call to someone who can teach you, and then you can be taught in your house! In fact, almost everything you can think about having a home-teaching market. (Even how to buy the merchandises :-)) With home-teaching, you do not need to go to the cram school of some classes that are full of people.

However, learning something by home-teaching costs a lot. You might need to pay a huge a amount of money for these "convenient services." In order to reduce the costs of home-teaching. A study group called "Just Teach Me!!" was founded. The main idea of it is using the "power of crowd." The members write down the subjects that they wanted to learn, and the subjects that they can teach (with the hourly wage they expected to have). Then the group will match the people according to the requirement.

Now, "Just Teach Me!!" provides a new option to the members. You can establish a sub-group for some members, the sub-group need to satisfy the rules that "every member in the sub-group is taught by exactly one member in the sub-group, and every member in the sub-group also teaches exactly one member in the sub-group."

If you use this new option, which is stated as above, then everybody in the sub-group are just need to pay the "average" amount of the hourly wages to the member who teaches you, and then get the same amount, say c, from the member you teach, theoretically – so you can learn free! ...No, because the member need to pay the handling fee to the"Just Teach Me!!" group. Since the fee needed to pay is proportional to c, so the members want to minimize c, in order to pay less handling fee.

Given the information registered by the members, please find the minimum c among all possibly sub-groups.

■ Input Format

The input consists of multiple test cases, the first line of input contains an integer $T (T \leq 30)$ which denotes the number of test cases.

For every test case, the first line has one integers $N (2 \leq N \leq 200)$ which indicates the total number of members in the"Just Teach Me!!" group. Then follows an $N \times N$

matrix A, with entry a_{ij} equal to the hourly wage which is needed to pay for the j-th member being taught by the i-th member ($0 \leq a_{ij} \leq 200000$). If $a_{ij} = -1$, then it means that the i-th member cannot teach the j-th member. ($a_{ii} = -1$ for all i)

■ Output Format

For each test case, please print a single line with a fraction in form p/q denote the minimum c among all possibly sub-groups. If there has no valid sub-group, then print "`Impossible`" (without quotes).

Notice that the greatest common divisor of p and q needs to be 1. (i.e., $\gcd(p, q) = 1$)

■ Sample Input

```
3
5
-1 5 5 5 5
5 -1 5 5 5
5 5 -1 5 5
5 5 5 -1 5
5 5 5 5 -1
6
-1 12 -1 8 -1 -1
-1 -1 9 -1 -1 -1
7 -1 -1 -1 -1 -1
-1 -1 -1 -1 -1 10
-1 -1 -1 10 -1 -1
-1 -1 -1 -1 10 -1
3
-1 10 15
-1 -1 20
-1 -1 -1
```

■ Sample Output

```
5/1
28/3
Impossible
```