Problem A
Prime Factors Sum
Time limit: 3 seconds

Given a positive integer n, let $f(n)$ be the sum of prime factors of n. Define $f^m(n)$ as:

$$f^1(n) = f(n) \text{ and } f^m(n) = f^{m-1}(f(n)) \text{ for } m \geq 2.$$

We are interested in the value of

$$g(n) = \lim_{m \to \infty} f^m(n).$$

Take $n = 63 = 3 \times 3 \times 7$ as an example. The prime factors of 63 are 3 and 7. Thus $f^3(63) = f^2(f(3 \times 3 \times 7)) = f^2(3 + 7) = f^2(10) = f(f(2 \times 5)) = f(2 + 5) = f(7) = 7$. Note that, by definition, $f(n) = n$ if n is a prime. Therefore $g(63) = 7$.

Input Format

The first line is an integer N $(1 \leq N \leq 10)$ which indicates the number of test cases. Each test case occupies one line and contains one positive integer n, $n \leq 4,000,000,000$.

Output Format

For each test case, output the number $g(n)$.

Sample Input

```
2
63
15
```

Sample Output for the Sample Input

```
7
2
```

Problem B
Compute Center of a Tree
Time limit: 1 seconds

Let $T = (V, E)$ be a tree with vertex set $V = \{1, 2, \ldots, n\}$ and edge set E. For example, the following is a tree with $n = 4$ vertexes.

```
  2--4
 / \
1   3
```

Let u and v be two vertexes in a tree T. The distance $d(u, v)$ is the number of edges from u to v in T. For example, in the above tree, $d(1, 4) = 2$, $d(2, 3) = 1$.

Let v be a vertex in a tree T. The *eccentricity* $\epsilon(v)$ of the vertex v is the maximum distance between v and any other vertex u of T. For example, in the above tree, $\epsilon(1) = 2$, $\epsilon(2) = 1$.

The center of a tree is the vertex v with minimum $\epsilon(v)$. Note that the center of a tree may not be unique. For example, the following tree with $n = 5$ vertexes has 2 centers, 1 and 4.

```
    4
   / \
  1   5
 / \
2   3
```

Write a program to compute the centers of a tree.

Input Format

Each test case consists of two lines. The first line contains a positive integers n. The value of n is less than 1000. The second line described the edges of the tree.

It is convenient to consider a tree T as a *rooted* tree. Thus, an edge can be described as a pointer to its root. Therefore, the second line of the input data for a tree contains n integers $p_1, p_2, \ldots, p_n$, where each p_i is the parent of vertex i in T. If $p_i = 0$, then vertex i is the root of the tree.

Note that the test data file may contain many test cases. The last test case is followed by a line containing a single 0.

Output Format

The outputs for each test case should be the centers of the tree. If the tree has more then 1 center, list the centers in ascending order.

Sample Input

```
4
2 0 2 2
5
4 1 1 0 4
0
```

Sample Output for the Sample Input

```
2
1 4
```

Problem C
Star Gate Exploration
Time limit: 1 seconds

In year 3000, humans discover a buried Star Gate first on a planet X. Some unknown alien race used the Star Gate to do intergalactic travel among planets. To dial the Star Gate, you need to enter an integer number. It turns out the integer number is a mysterious code representing the planet X, that is, not the destination you want to dial to. Once the worm hole is created, you have only one pre-built destination to go to.

After the discovery on planet X, humans discover more Star Gates on other planets. In the beginning, humans send explorers to travel through Star Gates. Eventually, humans find these Star Gates trap the explorers in a cycle. Instead of sending human explorers, robots will be sent to find the cycle. These cycles are crucial for understand the alien civilization. Given a Star Gate and a Star Gate system, your task is to write a program for the robot to collect these cycles. Note that a Star Gate can send a robot back to itself; that is, it is a self-loop. A self-loop is a cycle.

Input Format

The test data begins with a number $N(\leq 1000)$ which is the number of star gates. Following the number N is N line of star gate information represented by two integers $(S\ D)$ where S is the index of a source planet and D is the index of destination planet. $0 \leq S, D \leq 9999$.

Following the star gate information is a number M which is the number of robots to be sent. Following the integer M is M lines of planet indexes which are the starting planets for these robots to begin.

Output Format

For each robot, please print the cycle. A cycle is composed of looped planet indexes separated by a space.

Sample Input

```
9
0 6
1 6
2 0
```

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

Sample Output for the Sample Input

```
6 3 1
4
```

Problem D
Marginal Planet
Time limit: 3 seconds

There is a marginal planet in the universe called as Marginal Planet. People on the Marginal Planet can live almost forever in ordinary conditions but will die without friends. The most important element supporting their life is connection with friends. If some person has at most f friends, he will be lonely and frustrated, and will die in one year. This effect is call marginal effect and such kind of person is called marginal group. (For example, $f = 0$ means that person in marginal group has no friends.) Initially, there are many persons living on the Marginal Planet with no marginal group. But one day, a comet from a deep area of the universe strongly hit the Marginal Planet. Many persons dies on this disaster, and some persons become a new marginal group. Marginal effect happens. At the end of first year, persons in the marginal group die, and some persons happen to form a new marginal group and die at the end of second year and so on. The marginal effect will always take place until there is no marginal groups. Assume that there are n persons $A_1, A_2, A_3, \ldots$, and A_n living on the Marginal Planet. After the disaster, friendship connections between persons are that person A_u and person A_v are friends if $u + v$ is one of some numbers $B_1, B_2, B_3, \ldots$, and B_m. Given a number p, after how many years will person A_p die due to marginal effect?

Take $n = 7, p = 3, f = 1, m = 3, B_1 = 7, B_2 = 8$, and $B_3 = 12$ as an example. That is, there are 7 persons $A_1, A_2, A_3, \ldots$, and A_7 living on the Marginal Planet. After the disaster, friendship connections are unordered pairs as following $(A_1, A_6), (A_2, A_5), (A_3, A_4), (A_1, A_7), (A_2, A_6), (A_3, A_5)$, and (A_5, A_7). Remark that self-connection (A_4, A_4) and (A_6, A_6) are out of consideration. At first year, person A_4 has only one friend A_3 and dies at the end of first year, and then A_3 happens to form a new marginal group and die at the end of the second year. Finally, connections between A_1, A_2, A_5, A_6, and A_7 forms a cycle. Each one of them has exactly 2 friends and survives forever. So after 2 years, person A_3 will die due to marginal effect.

Input Format

The first line is an integer N $(1 \leq N \leq 10)$ which indicates the number of test cases. Each test case occupies a line and contains integers n $(1 \leq n \leq 100,000)$, p $(1 \leq p \leq n)$, f $(1 \leq f \leq 5)$ and m $(1 \leq m \leq 6)$ followed by $B_1, B_2, B_3 \ldots$, and B_m $(1 \leq B_i \leq 200,000)$.

Output Format

For each test case, output a number Y that person A_p will die at the end of Y years due to marginal effect. If person A_p will survive forever, output -1.

Sample Input

```
2
7 3 1 3 7 8 12
7 5 1 3 7 8 12
```

Sample Output for the Sample Input

```
2
-1
```

Problem E
Cao-Chong Weighed the Elephant
Time limit: 3 seconds

The lord of Wei, Cao-Cao, had a very clever son, Cao-Chong. Cao-Chong was so bright that he was as smart as an adult when he was only five or six. One day, the lord of Wu, Sun-Quan, sent Cao-Cao an elephant as a present. Cao-Cao could not figure out the weight of the elephant by questioning all the servants there. None of them could think of a good way to weigh the elephant because there was not such a huge scale for it. Cao-Chong suggested that the servants led the elephant into the ship and recorded the water level of the ships side. Then he told the servants to walk the elephant out of the ship and put a great number of rocks which made the ship reach the same water level recorded earlier. Finally, the servants were told to scale the rocks in batches then added up all the weight of the rocks, which revealed the weight of the elephant. Cao-Cao was very happy and decided to weigh elephants in the same way. Now there are n kinds of rocks with different weights. The weight of each rock is $R_1, R_2 \ldots, R_n$ where $1 \leq n \leq 100$ and $1 \leq R_i \leq 100000$. The number of each kind of rock is $A_1, A_2, \ldots A_n$ where $1 \leq A_i \leq 1000$. The known information is that the weight of an elephant is less than M, $1 \leq M \leq 100000$. Therefore, how many different weights can people precisely measure between 1 and M, including 1 and M? For example, if there are two kinds of rocks, and the elephant's weight is no more than 5. There are 2 of rocks with weight 1 and one rock with weight 4. Then we will be able to weigh the elephant for weight 1, 2, 4, and 5, so the answer is 4.

Input Format

An instance of the problem consists of two lines. The first line has two integers n ($1 \leq n \leq 100$) and M ($1 \leq M \leq 100000$) indicate there are n kinds of rocks and the elephant weights no more than M. The second line has $2n$ integers which are shown in $R_1, R_2 \ldots, R_n$ and $A_1, A_2, \ldots A_n$ where $1 \leq R_i \leq 100000$ and $1 \leq A_i \leq 1000$. Note that the test data file may contain more than one instances. The last instance is followed by a line containing two zeros.

Output Format

For each test data, output the number of different weights that may be weighted by using the given rocks in one line.

Sample Input

```
3 10
1 2 4 2 1 1
2 5
1 4 2 1
0 0
```

Sample Output for the Sample Input

```
8
4
```

Problem F
Amusement Park
Time limit: 2 seconds

The long awaited NCPC amusement park is finally opening this fall. At anytime, at most R different rides will be in service. The rides in the park are all on point system, meaning that certain number of points is required to take a particular ride. Rider must have enough points in his/her park pass in order to take a ride.

To celebrate the grand opening of the park, free passes (with points) are being given away in a lottery. Lucy is a lucky person to win one of the lotteries. She received a pass with n points. She goes to the park to collect her pass and decides to look around and maybe take some rides. However, she does not need to use up all the points at once. She can take 0 or more rides and save the points for future visits to the park.

Please write a program to help determine in how many different ways can Lucy use her points in her first visit to the park?

Technical Specification

1. At most R rides are available on the day of Lucy's first visit to the park, $1 \leq R \leq 40$.
2. Lucy's winning pass has P points, $1 \leq P \leq 10^{18}$.
3. The points required for each ride is at most 10^{16}.
4. Lucy will take each ride at most once during her first visit.

Input Format

An instance of the problem consists of several lines of input. The first line contains two integers, R and P, denoting the number of rides available and the points in Lucy's pass, respectively. The next line contains R integers separated by a space, denoting the points required for each of the R rides. The last instance is followed by a line containing two 0's.

Output Format

For each instance, output on a single line the total different number of ways for Lucy to use her points in her first visit.

Sample Input

```
3 100
10 50 100
5 1000
50 100 100 1001 500
0 0
```

Sample Output for the Sample Input

```
5
16
```

Note that the first output
5
means 0 rides, or rides with 10, 50, 100, 10+50 points on first visit.

The second output is
16
Assume first 100 in input is 100A and second 100 is 100B, can take 0 rides, or rides with 50, 100A, 100B, 500, 50+100A, 50+100B, 50+500, 100A+100B, 100A+500, 100B+500, 50+100A+100B, 50+100A+500, 50+100B+500, 100A+100B+500, 50+100A+100B+500 points on first visit.

Problem G
Optical Character Recognition
Time limit: 5 seconds

Optical Character Recognition (OCR) is a problem to translate an image that contains English characters into ASCII characters. This problem in general is very difficult considering the variety of fonts and sizes. However, in some application domains, the images are captured from screen, which are clean and without rotations. The characters are printed from a so-called true-type font.

Let the size of the printed characters be known before the recognition. You are ordered to build a simple OCR system by template matching. Template matching (TM) is a basic technique in digital image processing. Given a *template image* (often a smaller region of image), the goal is to find the locations of a 2D image (often a larger image to be searched, called *search image*) which match a template image.

We will call the search image $S(r, c)$, where (r, c) represents the coordinates (i.e., row# and column#) of each pixel in the search image. We will call the template $T(r_t, c_t)$, where (r_t, c_t) represents the coordinates of each pixel in the template. We then simply move the center (or the origin) of the template $T(r_t, c_t)$ over each (r, c) point in the search image and calculate some differences over the whole area spanned by the template. The position with the score lower than a given threshold is considered a match in the search image. Let the images to be searched be binary; that is, 0 or 1. Let the differences between $S(r, c)$ and $T(r_t, c_t)$ be SAD (sum of absolute differences). The definition of SAD is as follows:

Let the pixel value in the search image with coordinates (r_s, c_s) is $I_s(r_s, c_s)$, which is 0 or 1. A pixel in the template image with coordinates (r_t, c_t) has value $I_t(r_t, c_t)$, which is 0 or 1. Thus the absolute difference in the pixel value is defined as $Diff(r_s, c_s, r_t, c_t) = |I_s(r_s, c_s) - I_t(r_t, c_t)|$

$$SAD(r, c) = \sum_{i=0}^{T_{rows}-1} \sum_{j=0}^{T_{cols}-1} Diff(r + i, c + j, i, j)$$

Where T_{rows} and T_{cols} denote the number of rows and the columns of the template image. By looping through the pixels in the search image, some positions with low SAD scores give the estimate for some best positions of template within search image. Let the number of pixels of a template image is N_t. A suggested threshold V can be computed by:

$$\frac{N_t - SAD(r, c)}{N_t}$$

For example, you can set $V \geq 0.9$ as a condition to report a match.

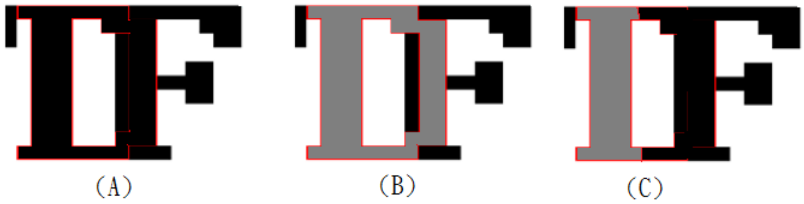

Figure 1: The false-negative examples.

Figure 2: Remove the pixels of a matched template.

Since the size of the printed text in the image is given, it seems we can use TM to recognize the printed characters straightforwardly. Unfortunately, it is not. For example, in Figure 1.(A), it is very obvious to human that the two characters are 'T' and 'F'. However, their pixels can be connected after printed on screen. So, if no precise threshold is set, 'D' can be recognized as well (see Fig. 1.(B)). If a template of 'I' is matched before 'T', Fig. 1.(C) shows that an 'I' is recognized. In practice, adjusting the threshold values to an optimal solution is mission impossible. This is a well-known nightmare to image recognition systems.

One day, a clever idea comes to your mind. Once a template image satisfies a given threshold, you can remove the black pixels of template image like Fig. 2. An interesting fact is that if you have a precise matched, the remaining black pixels are minimum. Fig. 3(a) is a case of best match which leaves minimum number of black pixels. On the other hand, Fig. 3(b) is a case of poor match.

Given a set of template images and search images, please print the translated characters from your image recognition program. Hint: basically you still need a threshold for template matching. For example $V = 0.9$ is a good enough threshold for your TM to tolerate some noises. However, if you solve the problem successfully, the precision of the threshold value is no longer crucial.

Input Format

The test data begins with a number $N(\leq 62)$ where N is the number of template images (52 letters and 10 digits). The second line of test data begins with template images. Each template image begins with a letter from 'a'-'z','A'-'Z', or '0'-'9' which is the ASCII character in the template image. Next, two integers ($row\#$ $col\#$) are given to define the dimension of a template image. The pixels of a template image is described by a matrix of '*' and '.', where '*' stand for a black pixels and '.' stands for a white pixel.

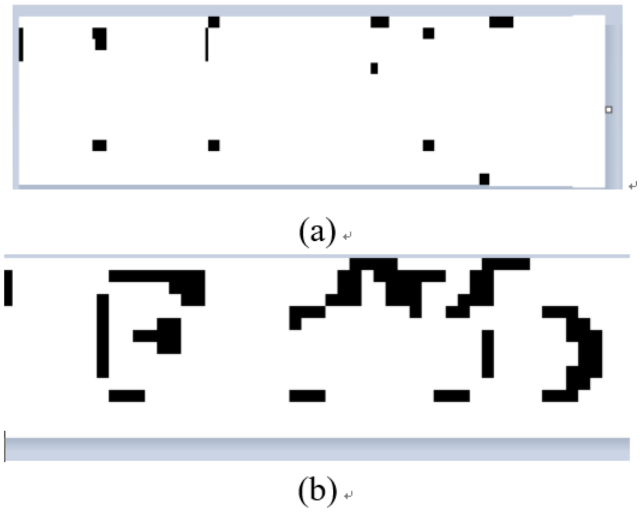

Figure 3: (a) A best match where remaining black pixels are minimum. (b) A poor match where a lot of black pixels remain.

Following the template images is a number S, which is the number of search images. Each search image is defined by two integers ($row\#$ $col\#$). The pixels are described in the same way as template images.

Output Format

Please output the translated ASCII characters in each search image from left to right in a line.

Sample Input

```
3
L
5 3
*..
*..
*..
*..
***
I
5 3
.*.
.*.
.*.
.*.
.*.
U
```

```
5 5
*...*
*...*
*...*
*...*
*****
1
5 20
...*..*...*...*.....
...*..*...*...*.....
...*..*...*...*.....
...*..*...*...*.....
...********...*.....
```

Sample Output for the Sample Input

```
LUI
```

Problem H
The Sum and Product of Eigenvalues
Time limit: 2 seconds

The eigenvalues of a real matrix $A \in R^{n\times n}$ play an important role of a variety of applications. Let $\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_n$ be eigenvalues of a real matrix $A \in R^{n\times n}$, denote the sum and product of eigenvalues of $A \in R^{n\times n}$ as

$$\begin{aligned} S &= \sum_{i=1}^{n} \lambda_i &&= \lambda_1 + \lambda_2 + \cdots + \lambda_n \\ P &= \Pi_{j=1}^{n} \lambda_j &&= \lambda_1 \times \lambda_2 \times \cdots \times \lambda_n \end{aligned}$$

For a real square matrix $A \in R^{n\times n}$, the sum S and product P of its eigenvalues must be real even though there is no guarantee that all of $\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_n$ are real.

This problem asks you to write a program to compute the sum S and product P of the eigenvalues $\lambda_1,\ \lambda_2,\ \ldots,\ \lambda_n$ for a real matrix $A \in R^{n\times n}$.

Restrictions: Each test matrix A has dimension up to 8, and its elements are all integers in the range [-16, 15].

Input Format

The first line of the input file contains one integer, $K \leq 5$, indicating the number of test cases to come. Each test data set consists of $n+1$ lines, where the first line is an integer n which indicates the dimension of the matrix, followed by n lines with each line having n numbers separated by at least one space or a tab.

Output Format

For each data case, report the sum of eigenvalues S followed by the product of eigenvalues P for each given real matrix by round-off up to an integer in a single line where S and P are separated by a tab or spaces.

Sample Input

```
2
3
 -2 -1  0
  1 -2  0
  0  0  3
6
 1  0  0  0  0  0
 2 -1  0  0  0  0
 0  0  2  0  0  0
 0  0  3 -2  0  0
 0  0  0  0  3  0
 0  0  0  0  5 -3
```

Sample Output for the Sample Input

```
 -1  15
  0 -36
```

Problem I
Compute Territory
Time limit: 3 seconds

Problem Description

In a city, there are some convenience stores. A resident in the city usually goes to the convenience store closest to his (or her) location. Therefore, the stores inside the city divide the city into territories where each store has its own territory and any point inside will find it to be the closest store in the city. For ease of discussion, the city is assumed to be rectangular, and we only consider territories inside the city. Consider the example in Figure 1. The city has length 8 and width 12, and there are six stores in the city, represented by S_1, S_2, S_3, S_4, S_5, S_6, whose coordinates are (2, 2), (6, 2), (10, 2), (2, 6), (6, 6), (10, 6), respectively. These stores divide the city into six territories (see Figure 1).

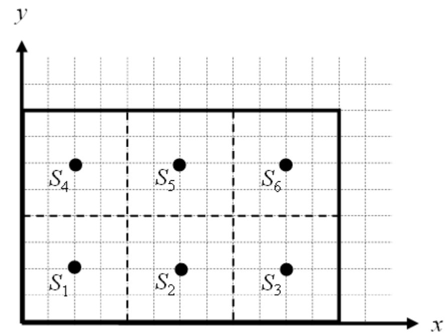

Figure 1: An illustration.

Given a city and the existing stores in it, we wish to open a new store in the city and compute the territory it would establish. For instance, given the city and stores in Figure 1, by opening a store S' at coordinate (8, 8), the new store will establish a territory of 8 units (see Figure 2).

In this problem, you are given the dimensions of a city, the coordinates of N stores $S_1, S_2, \ldots, S_N$, and the coordinate of a new store S' (every store has a different coordinate and is inside the city). Please compute the territory the new store S' will establish in the city, and output the area of the territory. The output should be rounded to the nearest integer.

Technical Specification

- The number of test cases is at most 10.

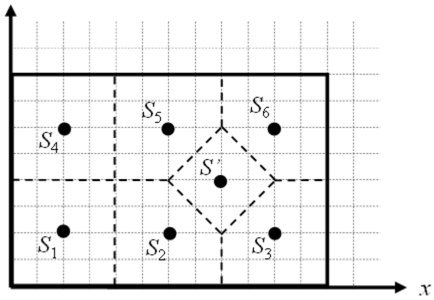

Figure 2: An illustration.

- The number N of stores is an integer between 0 and 100.
- The length l and width w of the city are integers between 1 and 1000000.
- The x-coordinate of a store is a floating point number between 0 and $w - 1$.
- The y-coordinate of a store is a floating point number between 0 and $l - 1$.

Input Format

The first line of each test case gives 3 integers l, w, N, and two floating point numbers x', y' ($1 \leq l$, $w \leq 100000$, $0 \leq x' \leq w - 1$, $0 \leq y' \leq l - 1$, and $0 \leq N \leq 100$), where l and w represent the length and width of the city, respectively, N represents the number of existing stores in the city, and (x', y') is the coordinate of the new store. The first line is followed by N lines each of which gives the x- and y-coordinates of an existing store ($0 \leq x \leq w-1$ and $0 \leq y \leq l-1$). The input is terminated by a line containing five zeros, which should not be processed. No floating point number has more than 2 digits after the decimal point.

Output Format

For each case, output the area of the territory established by the new store. The output value should be rounded to the nearest integer.

Sample Input

```
8 12 6 8 4
2 2
6 2
10 2
2 6
6 6
```

```
10 6
8 12 5 8.5 4.5
2.5 2.5
6.5 2.5
10.5 2.5
2.5 6.5
6.5 6.5
0 0 0 0 0
```

Sample Output

```
8
16
```

Problem J
Temperature Overheating Alarming Technique
Time limit: 5 seconds

For a quasi-stationary System-on-Chip(SoC) platform, the system temperature increasing trend is an approximated exponential curve, which can be describe as

$$T(t) = T_{ss} - (T_{ss} - T_0) \cdot e^{-bt}.$$

The T_{ss} is the temperature in the static state, which defines the temperature of the system is stable. The T_0 represent the initial temperature of this system, which is usually the ambient temperature. The b is a physic constant, which is depends on the process technology. At last, the $T(t)$ is the temperature at time t. In the practical way, there is one embedded thermal management unit (TMU)and one thermal sensor on the chip to prevent the SoC from overheating. When the chip temperature is higher than the alarming level, the TMU will stop the data process until the chip temperature is low enough. We assume the b is equal to 1.98. Besides, the embedded thermal sensor provides the sensing temperature every Δt_s, and the Δt_s is equal to 0.01 second.

Given a temperature tracing data consisting of three historical temperature data, $T(t-2\Delta t_s)$,$T(t-\Delta t_s)$ and $T(t)$. You need to find a way to predict when the chip temperature will be above the hard thermal limit, which is defined as the upper bound of the valid chip processing temperature.In this problem, we assume that the TMU will be turned off in your prediction. It means that the system temperature will be increased when you start to predict the future temperature.

Input Format

A temperature tracing data set consists of three temperature data, which are $T(t-2\Delta t_s)$,$T(t-\Delta t_s)$ and $T(t)$ respectively.The $T(t)$ is the current temperature and $T(t-2\Delta t_s)$,$T(t-\Delta t_s)$ are the temperature before $2\Delta t_s$ and Δt_s respectively. The temperature data all are positive floating-point numbers in the range $[50, 100]$. Note that the test data file may contain more than one temperature tracing data set and the TMU may affect the temperature increasing tread in the given temperature data set (*i. e.,* the temperature may decrease). The last data set is followed by a line containing a single 0.

Output Format

The output for each test case is the prediction result of when the chip temperature will be above $100°C$. Note that Δt_s is equal to 0.01 second, and the output unit is second.

Sample Input

```
92.5 85.1 90.1
92.5 97.5 90.1
0
```

Sample Output for the Sample Input

```
0.030
0.020
```

Problem K
Computer Cluster
Time limit: 5 seconds

An engineer, Raymond, works in Networking Communication and Processing Center (NCPC). His boss asked him to construct a computer cluster from a set of n computers (nodes). For convenience, these n nodes are represented by $\{1, 2, \ldots, n\}$. This computer cluster consists of a set of tightly connected computers (nodes) that work together so that, in many respects, they can be viewed as a single system.

To construct the preferred computer cluster, when connecting two nodes u and v using a communication link (bidirectional), it takes cost $w(u, v)$, where $w(u, v)$ is a positive integer. Moreover, the cost function w satisfies the following properties:

1. $w(u, u) = 0$;
2. $w(u, v) = w(v, u)$;
3. $w(u, v) \leq \frac{3}{5} \cdot (w(u, x) + w(x, v))$ for any three nodes u, v, x, for example, $w(u, v) = 7$, $w(u, x) = 8$, and $w(x, v) = 9$.

Raymond figures out an efficient algorithm to construct a special tree structure as the preferred computer cluster T. He selects a specified node $c \in \{1, 2, \ldots, n\}$ called *central hub* and select another distinct p nodes $u_1, u_2, \ldots, u_p$, called *hubs*, as the p children of c. Note that there are $n - p - 1$ non-hub nodes remained. For each non-hub node v, the algorithm finds exactly one u_i for some $1 \leq i \leq p$ as the *parent* of v, that is, setting a communication link between u_i and v.

In order to measure the performance of a computer cluster T, define $d_T(u, v)$ to be the distance between u, v in T (i.e., the sum of the costs of the edges in the path between u and v in T), and define the *diameter* of T as $D(T) = \max_{u,v \in T} d_T(u, v)$.

Given a central hub c and a positive integer p, the task of Raymond is to construct a depth-2 spanning tree T^* rooted at c (central hub) such that c has exactly p children (hubs) and the diameter of T, $D(T^*)$, is minimized. Your task is to write a computer program to help Raymond to compute $D(T^*)$.

Technical Specification

1. $2 \leq n \leq 2500$.
2. The cost of each edge $w(u, v)$ is a positive integer such that $1 \leq w(u, v) \leq 10000$.
3. $1 \leq p \leq \frac{n-1}{2}$

Input Format

The first line of the input file contains an integer, denoting the number of test cases to follow. For each case, the first line contains three positive integers, n, c, and p, separated by a space. In the following $\frac{n(n-1)}{2}$ lines, each line contains three positive integers and any two consecutive integers are separated by a space. The first two integers denote two nodes and the third positive integer indicates the cost of setting a communication link between them.

Output Format

For each case, output the corresponding diameter.

Sample Input

```
1
4 1 2
1 2 7
1 3 9
1 4 7
2 3 8
2 4 7
3 4 8
```

Sample Output for the Sample Input

```
22
```

Problem L
Median Maximum Bandwidth
Time limit: 3 seconds

Joe works at a Network Service Company. His major duty is design efficient network architectures. More specifically, consider N network devices and M links, where each link connects two different network devices with a predefined bandwidth for both directions. For convenience, the network devices are numbered from 1 to N. Each device can fully utilize the bandwidths of the links connecting to it. Meanwhile, it must follow the flow conservation principle, i.e. except the source and destination devices, each device has the same total in-flow as the total out-flow.

To simplify the study, Joe ignores the possible delay of the devices. Joe is interested in finding out the maximum bandwidth (, which is known as the maximum flow in optimization theory) that can be achieved between any two network devices. There are $\binom{N}{2}$ possible pairs of source and destination and each pair has a maximum possible transmission bandwidth. Joe wants to find out the the median of the distinct maximum bandwidth. If there are K elements in increasing order, here the median occurs exclusively at the $\lfloor \frac{K+1}{2} \rfloor$-th smallest element. For example, suppose there are 3 network devices labeled from 1 to 3 and there are 2 links that connect device 1 and device 2, and device 2 and device 3, with bandwidth 3 and 1, respectively. Then it is clear that the bandwidth between device 1 and device 3 is 1. Since there are 2 distinct values, 1 and 3, the median is 1.

Note that different pair of source and destination may have the same maximum bandwidth. Your task is to write a program to help Joe find out the median of the distinct maximum transmission bandwidths.

Input Format

The first line of the input gives the number of test cases, T ($T \leq 6$). T test cases follow. For each case, the first line consists of two positive integers N ($N < 1000$) and M ($M = O(N)$), separated by space(s), indicating the number of network devices and links, respectively. Then M lines follow, where each has 3 positive integers: I, J, B, separated by space(s), indicating the bandwidth between device I and device J is B ($B \leq 100$).

Output Format

The output is the median of distinct maximum bandwidths. For each test case, output the answer in a separate line.

Sample Input

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

Sample Output for the Sample Input

```
1
2
```