ACM-ICPC Asia Phuket Regional Contest
Phuket, Thailand

Practice Session
3 November 2011

You have 60 minutes and **4** problems to solve.

Problem A. Reverse and Sort

Given a set of positive integers, write a program to reverse the digits of each number and then sort them in ascending order. Note that any leading zeros are omitted.

Input
The first number is n $(1 \le n \le 10^6)$ which determines the number of positive integers to be processed, followed by n integers. A line may contain any number of integers. The value of input integer does not exceed 10^{12}.

Output
The output contains n lines of sorted numbers according to the process described above.

Sample input	Sample output

`10 5 2233`
`1601 90100 13009 802`
`50000000`
`301 7654321`
`210`
	`5`
	`5`
	`12`
	`103`
	`109`
	`208`
	`1061`
	`3322`
	`90031`
	`1234567`

Problem B. Line of Best Fit

Finding a line of best fit for a set of data is one of fundamental problems in statistics and numerical analysis. The problem can be formulated as follows. Suppose our data consists of a set P of n points in a plane, denoted $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)$. Given a line L defined by the linear equation $y = ax + b$, we say that the error of L with respect to P is the sum of its squared "distance" to the points in P:

$$Error(L, P) = \sum_{i=1}^{n} (y_i - ax_i - b)^2$$

The *least square approach* is to find the line L which minimizes such error. Your task is to write a program to find the values a and b of the line L for a given set of points P.

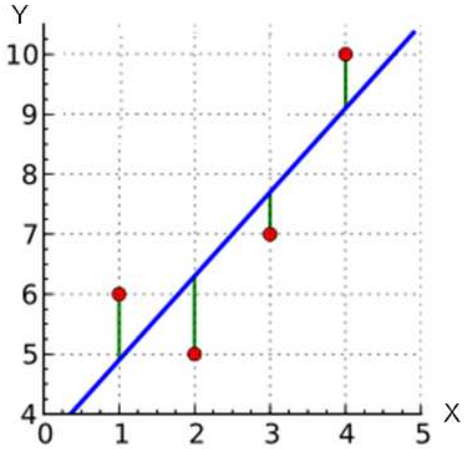

Figure 1. Example of points and a line of best fit

Input

The first line contains a positive integer n $(1 \le n \le 1{,}000)$. The next n lines contain 2 integers: x_i and y_i in each line. $|x| \le 10^6$ and $|y| \le 10^6$

Output

The output contains the values ***a*** and ***b*** in **two lines** respectively. The numbers must be rounded to 3 decimal places.

Sample input	**Sample output**

`4`
`1 6`
`2 5`
`3 7`
`4 10`
	`1.400`
	`3.500`

Problem C. Software Licenses

You are starting a security company that needs to obtain licenses for n different pieces of cryptographic software. Due to regulations, you can only obtain these licenses at the rate of at most one per month.

Currently the license i is selling for a price of P_i dollars. However, they are all becoming more expensive according to exponential growth curves: in particular, the cost of license i increases by a factor of $R_i > 1$. This means that if license i is purchased t months from now, it will cost $P_i * (R_i)^t$ dollars.

Given a set of n license prices and their growth factors, you have to plan the purchase of these licenses within n months in order to minimize the total cost.

For example, you need to buy licenses A, B, C, D. Their prices and factors are the following.

$P_A = 200, R_A = 1.01$, $P_B = 300, R_B = 1.12$, $P_C = 400, R_C = 1.05$, $P_D = 650, R_D = 1.10$

You can buy all needed licenses within 4 months. To obtain the minimum cost of 1,633.06 dollars you have to purchase D in the first month (with its initial price), then B in the second month, and then C in the third month and A in the last month.

Input

The first line contains a positive integer n $(1 \le n \le 100)$ indicating the number of licenses you need. The next n lines contain 2 numbers: P_j and R_j in each line.

Output

The output is the minimum cost that you have to pay. The number must be rounded to 2 decimal places.

Sample input	Sample output
4 200.0 1.01 300 1.12 400 1.05 650 1.1	1633.06

Problem D
Crucial Links
(1 second, 32 MB)

You would like to broadcast the ACM ICPC Regional Contest event in Phuket in real time. To do so, you have to transmit the video data from the recording machine to the broadcasting server in Bangkok. You have the map of the entire network of Thailand. There are N routers and M one-directional links. Each link i, for 1 <= i <= M, can transmit data from Router F_i to Router T_i, with bandwidth B_i. It is possible that there are more than one link that transmits data from a particular router to another particular router.

The recording machine is connected to Router 1 with an unlimited-bandwidth link, and the broadcasting server is connected to Router N also with an unlimited-bandwidth link.

To transmit video data from Router 1 to Router N, you split the video data and transmit them using many transmission paths. A transmission path is a path from Router 1 to Router N. Each path can transmit data with some bandwidth b and utilize bandwidth b from every link used by this path.

It is possible to use many transmission paths with various bandwidths and many paths can share a link. However, the total bandwidth of all paths using any link i must not exceed the bandwidth B_i. The total video bandwidth is the sum of all transmission bandwidths of all transmission paths.

Given the network map, you can calculate the best possible scheme to maximize the transmission bandwidth from Router 1 to Router N. However, you are afraid that some link might fail to transmit data as fast as the claimed bandwidth. For some link i, it is still possible to obtain the best transmission bandwidths even when the link bandwidth drops from B_i to $B_i - 1$. But for some link i, if this occurs, the best video transmission bandwidth also drops; call this link *a crucial link*. Note that when considering if an edge i is crucial, we only consider only edge i, while assuming that all bandwidths of all other links remain unchanged. i.e., we consider each edge separately.

Your task
For each test case, write a program that compute the number of crucial links.

Input

The first line of the input contains an integer K, the number of test cases (1 <= K <= 15). Each test case is in the following format.

The first line of each test case contains a pair of integers N and M (2 <= N <= 300; 2 <= M <= 5,000). The next M lines contain link information. For 1 <= i <= M, line 1 + i contains 3 integers F_i, T_i, and B_i (1 <= F_i <= N; 1 <= T_i <= N; 1 <= B_i <= 1,000). The sum of all bandwidths B_i is at most 20,000.

Output

Your program must output K integers, each on a separate line. For each test case, your program must output the number of crucial links.

<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td><pre>3
2 3
1 2 10
1 2 5
1 2 7
4 3
1 2 10
2 3 5
3 4 6
5 7
1 2 2
1 3 3
2 3 10
3 2 10
3 4 4
2 4 2
4 5 5</pre></td><td><pre>3
1
3</pre></td></tr>
</table>