第七届广东省大学生程序设计竞赛

（GDCPC2009）暨第十二届中山大学程序设计竞赛(ZSUCPC2009)

Problem A

Airport Connecting Management

(Input File: a.in / Standard Output)

There is a very beautiful country called ICPC (International Cleanest and Prettiest Country). It's really very clean and pretty, but not so convenient in traffic. This year, you are hired to take the following task called ACM (Airport Connecting Management):

There are N cities in ICPC. Each city has exactly one airport. At first, there are no flights between them. You are asked to connect these airports by adding flights. The schedule must follow these rules:

1. Each flight is a two-way service that directly connects two airports.
2. No pair of airports can be connected by more than one flight.
3. From a city, you can go to every other city by at most two flights (go directly or transfer by an intermediate city).

Here comes the problem: What's the minimum number of flights you need to add to obey these rules?

Input:

There are multiple test cases in this problem.

The first line contains an integer T ($1 \le T \le 10$) telling the number of test cases.

Each test case is an integer N ($2 \le N \le 50$) in a line indicating the number of cities.

Output:

For each test case, output an integer telling the minimum number of flights in a line.

Sample Input:

```
2
2
3
```

Sample Output:

```
1
2
```

Problem B

Personnel Scheduling

(Input File: b.in / Standard Output)

Personnel scheduling has been widely studied for many years. The scheduling can be thought of as the problem of assigning employees to shifts or duties over a scheduling period so that certain constraints are satisfied or the assignment achieved some objective.

In this problem, we focus on a basic personnel scheduling system. Suppose there is a small workshop which manufactures accessory for some special tools. This time, the workshop received a series of order forms in a day. Each order form gives out a period of time. For an order form, the workshop owner has to recruit one laborer to manipulate the machine. In this period, other order forms cannot distract the laborer. But when the laborer finished his current work, he can take over another order form.

To minimize the cost, the workshop owner wants to recruit least amount of the laborers. Besides, he has to consider the work-time balance among the laborers --- the deviation of the work-time of each laborer should also be minimal. The definition of the deviation is that, suppose there are laborers and their work-times are $w_1, w_2, ..., w_m$, then $D = \sqrt{\frac{1}{m}[(\cdot} \overline{w} = \frac{1}{m}(w_1 + w_2 + ... + w_m) \overline{(w_m - \overline{w})^2]}$, where .

Your task is to work out the minimum number of laborers to be recruited and the minimum deviation between these laborers.

Input:

The first line of the input gives out the number T (T≤100) of test case. For each test case, the first line is an integer N (1≤N≤20), representing the number of the order form. The following N lines each line contains two timestamps, “start-time” and “end-time”. The format of a time forms “HH:MM” (0≤HH<24, 0≤MM<60). A laborer should be assigned to this period of time, start-time and end-time *inclusive*.

Output:

For each test case, output a line with two number *m* and *D*. *D* is rounded to 2 decimal places.

Sample input:

```
2
2
01:00 02:00
02:00 03:00
5
00:00 00:00
15:13 15:58
03:38 04:42
03:15 13:56
13:03 21:50
```

Sample output:

```
2 0.00
2 47.00
```

Problem C
Brick by Brick Game
(Input File: c.in / Standard Output)

Brick by Brick Game is a kind of creative building game. Like all toy bricks games, it uses a set of bricks to construct buildings.

In this problem, we consider a set of bricks which forms a building in rectangle shape. Our task is to remove K bricks from the building. In order to make the remaining part of the building keep standing, the following rules should be obeyed:

1. Each floor of the building (remaining part) consists of a single connected row of bricks.
2. At least one brick from each floor must rest upon some brick from the floor below it.

The only exception is the bottom floor, which rest upon ground.

3. At least one brick of the bottom floor must be left.

Each brick has its own visual value (a non-negative integer printed on the surface). We want to maximize the sum of the visual values on the bricks left. Can you tell me what the sum could be?

For detail, Figure 4.3.1 illustrated the case in the sample input. The maximized solution follows all of the above rules. Remember that, there's no restriction on how many floors that the remaining part of the building must keep, so we may abandon some top floors.

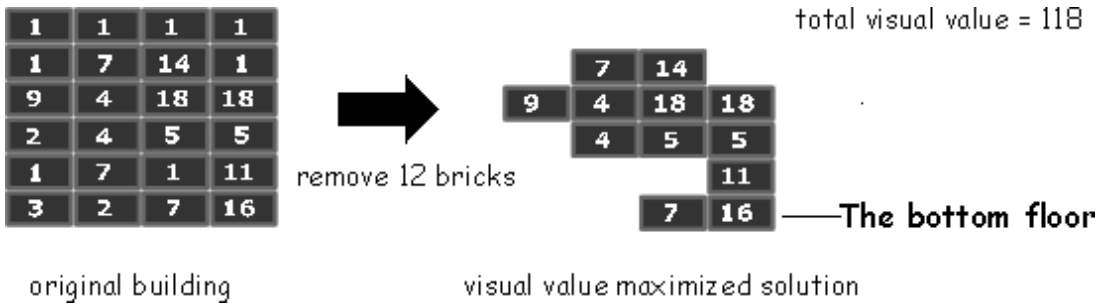


Figure 4.3.1 Example

Input:

The first line of the input contains a positive integer $T \leq 20$ specifying the number of test cases to follow.

The first line of each case contains three integers: N, M and K. ($0 < N, M \leq 64$, And $\max(N \times M - 64, 0) \leq K < N \times M$)

Next follow N lines, each with M integers specifying the visual values of each bricks. The first line describes the top floor of the building while the last line describes the bottom floor. All of the visual values are non-negative integers smaller than 32768.

Numbers are separated by spaces.

Output:

For each test case you should output a single line containing "Case X: Y" (quotes for clarity) where X is the number of the test case (starting at 1) and Y is the total visual value of the maximized solution.

Sample Input:

```
1
6 4 12
1 1 1 1
1 7 14 1
9 4 18 18
2 4 5 5
1 7 1 11
3 2 7 16
```

Sample Output:

```
Case 1: 118
```

Problem D

A Carnival Game

(Input File: d.in / Standard Output)

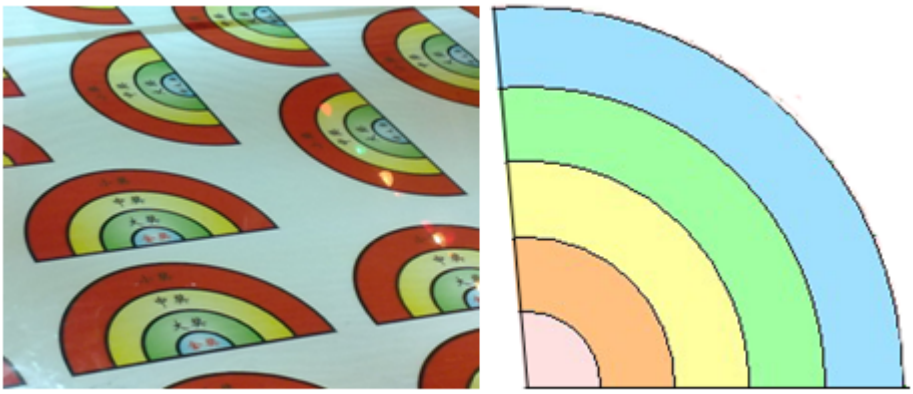
Maybe you have ever played such a kind of carnival game (If not, you can go to Tee Mall or Grandview Mall to have a try ^_^):

There are a lot of semicircles on the desk. Every semicircle is divided into several parts. You can throw a coin to the desk. If your coin lies in some part without intersecting any arcs or lines, you can get the corresponding prize, e.g., a small prize, middle prize, big prize or even

gold prize, as shown in Figure 4.4.1(a).

This problem is base on this game. To make things complicated, you have to deal with sectors instead of semicircles. There are N sectors on the desk. The sectors will not overlap. Every sector is divided into K parts. You can assume that the sector is cut from a group of concentric circles whose radius form an arithmetic progression (a list of numbers A_i satisfying that A_i-A_{i-1}=d for all $i > 1$). Initially, the whole desk is white. Then each part of every sector is colored differently. Figure 4.4.1(b) shows an example of a sector when K = 5.

Figure 4.4.2



(a) (b)

Example

You can throw some coins to the desk. A coin is a circle. If your coin lie in the colored part inside a sector and doesn't intersect any arc or segment of any sector, you can get some score. From the outer part to the inner part, the score you can get is 1, 2, 3, …, K separately. Otherwise, the score is 0.

Now, given the description of the N sectors and the Q coins, I'd like you to tell me the score every coin can get.

Note: A coin intersects an arc or a segment if and only if their minimum distance is less than 1E-6.

Input:

There are multiple test cases in this problem.

The first line contains an integer T ($1 \le T \le 10$) indicating the number of test cases.

For each test case, the first line contains two positive integers N and K ($1 \le N \le 5000$, $1 \le K \le 5000$), indicating the number of sectors and the number of parts.

Then N lines followed, each line contains five floating number x1, y1, x2, y2, r ($-1000000 < x1, y1, x2, y2 < 1000000$, $0 < r < 1000000$). It means that there is a sector whose arc is (x1, y1)-(x2, y2) (in counter-clockwise order) and radius is r. **The angle of the sector will never be more than 180 degrees.**

After that, there is a line containing one integer Q ($1 \le Q \le 20$), indicating the number of coins.

Then Q lines followed, each line contains three floating number x, y, r ($-1000000 < x, y < 1000000$, $0 < r < 1000000$). It means there is a coin whose center is (x, y) and radius is r.

Output:

For each test case, first print the case number (start from 1) in the format of "Case i:" in a line, as shown in the sample output.

Then print Q lines, each line contains an integer indicating the score of the corresponding coin.

Sample Input:

```
2
1 2
1 0 0 1 1
2
0.5 0.5 0.2
0.5 0.5 0.5
```

2 2
0.6 0 0 0.6 0.6
1.7 1 0.7 0 1
4
0.1 0.1 0.05
0.1 0.1 0.1
0.67 0 0.05
0.9 0.15 0.1

Sample Output:

Case 1:
1
0
Case 2:
2
0
0
1

Problem E

Publish or perish

(Input File: e.in / Standard Output)

Paper quality and quantity have long been used to measure a research's scientific productivity and scientific impact. Citation, which is the total times a paper has been cited, is a common means to judge importance of a paper. However, with all these factors varying, a collegiate committee has problems when judging which research is doing better. For this reason, H-index is proposed and now widely used to combine the above factors and give accurate judgment. H-index is defined as:

> A scientist has index h if h of [his] Np papers have at least h citations each, and the other $(Np - h)$ papers have at most h citations each.

In other words, a scholar with an index of h has published h papers each of which has been cited by others at least h times. Note that H-index is always an integer. It's said that achieving H-index of 18 means one is fully qualify to be a professor, and H-index of 45 or higher could mean membership in the United States National Academy of Sciences.

You are to calculate the H-index for all the researches in the list, base on the given information.

Input:

There are multiple scenarios in the input, terminated by a single zero (0).

Each of the scenarios begin with an integer N ($1 \le N \le 100$), means that there are N papers. N lines follow, each contain a string (not exceeding 20 characters long), representing the author of the corresponding paper, without white spaces in-between. Though it would be common for one paper written by several authors, there would be exactly one author of each of these papers. Finally, there are N lines of strings, containing '0's and '1's. If the j-th character in the i-th line is '1', it means the i-th paper cites the j-th paper. A paper could never cite itself.

Output:

For each scenario, output as many lines as the number of authors given. Each line contains the author's name and his H-index. The list should be sorted first by H-index in descending order, than by name in alphabetic order (Actually, ASCII order. So 'B' is prior to 'a').

Output a blank line after each scenario.

Sample Input:

4
Peter
Peter

Bob
Bob
0000
1000
1100
0100
0

Sample Output:

Peter 2
Bob 0

Problem F
Dreams in the East
(Input File: f.in / Standard Output)

As a game master, Edwin's job is to design maps for Dreams in the East, a famous online game produced by the company he has been working for, the NetHard. One of his favorite designs is the so-complex-road-system connecting Anchang City and Cunfang Hill in the game.

Unfortunately things have got somehow out of control recently, as Edwin found that some of those smartest players have already revealed the shortest path between Anchang City and Cunfang Hill. With such information being spread, players are now simply crowding on the shortest path and ignoring others, which makes the so-complex-road-system look like a so-naive-road-system.

Edwin is now aggravated and the consequence could be serious. However, while maps can be redesigned and become more complicated easily, such measures may not be accepted by most players with ease: changing of existing roads or adding too many roads could lead to a bunch of complaint, which is very likely to threat Edwin's job, especially during this damned financial crisis. Hence Edwin needs another way out, to put it simple, he wonders if he can add one road only in the map to fulfill his somewhat "unkind" wish to players.

So here comes the problem:

The map is described as N way points and M bidirectional roads. Way points are numbered from 1 to N, with 1 for Anchang City and N for Cunfang Hill. A path connecting Anchang City and Cunfang Hill will be described as a sequence of roads, and two paths are different if the two sequences describing them are different by at least one road.

Each road is weighted with the average time required to pass it, and a path will be considered the shortest if the sum of the average time needed to travel through it is minimal. Note, there could be many different shortest paths connecting the same pair of points.

To get the job done, Edwin needs to choose two distinct way points which are not connected directly by an existing road and then link them up with a new road. As to the weight, it is a positive integer determined sheerly at Edwin‘s will, because as a game master he can dispatch a tough monster to guard the way, or simply add a light-speed car to deliver a player in a flash. However Edwin's aim is to increase the number of different shortest paths by at least a given number D, and now he wonders how many different choices he has when choosing those two way points. By the way, choosing point A and B is considered the same as choosing B and A.

Input:

There are multiple test cases.

Each test case starts with an integer N indicating the number of way points, and an integer M indicating the number of roads.

Then a line contains the integer D which is the extra shortest path Edwin wants.

Then M lines follow. Each line consists of three positive integers A, B and C, indicating an edge connecting vertex A and B with weight C.

Input is terminated by a line containing two zero "0 0".

It is guaranteed that

$2 \le N \le 20000$

$0 \le M \le 100000$

$1 \le A, B \le N$

$1 \le C \le 10000$

$1 \le D \le 1000000000$

No two roads of a test case may connect a same pair of points, and no road connects a point to itself.

Output:

For each case just output the answer in one line.

Sample Input:

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

Sample Output:

```
1
1
1
35
```

Problem G

Black Jack

(Input File: g.in / Standard Output)

Netease Interactive Entertainment (NIE) is interesting in developing funny leisure games for users. Nowadays some engineers are doing research on a famous game called Black Jack. Here let's make the rules a little simpler. Considering that only the banker and you play the game with a number of cards valued form 1 to 10, and the game is consisted of several rounds; you and the banker play it round by round until there are less than 6 cards left. In each round of the game:

1. The banker and you must take two cards one by one in order, that means first the banker take one card, and then you take one, then the banker, and then you. After

that the banker and you take turns to make some ‘call’, the banker first.

2. When it’s one’s turn to make a ‘call’, he can choose to take one more card or not. But if one chose not to take the card, he can not make ‘call’ any more in this round.
3. When it’s the banker’s turn to make a ‘call’, if the sum of his cards is not greater than 16, he must take one more card. Or else he will not take the card.
4. If one’s sum of his cards is larger than 21, he loses and this round ends immediately.
5. When it’s your turn to make a ‘call’, if the banker can’t make ‘call’ any more and the sum of your cards is already larger than the banker’s, you must choose not to take the card and this round ends.
6. If both you and the banker choose not to take the card, or there is no card left, the banker wins this round if the sum of his card is not less than yours, otherwise you win this round.

Now given N cards in order, one of the most valuable information that the engineers want to know is the number of rounds that you can win at most, could you help them?

Input:

The first line of input will be a positive integer T≤15, T test cases follow.

For each case, there will be two lines. The first line will be a positive integer N≤1000, representing the number of cards. Then N positive integers follow the next line, each number will be larger than 0 and less than 11, representing the value of the given N cards.

Output:

For each test case, output an integer on a single line, the number of rounds that you can win at most.

Sample Input:

```
2
6
10 10 7 3 2 3
8
1 1 1 1 10 10 1 1
```

Sample Output:

```
1
1
```

Problem H

Westward Journey Online

(Input File: h.in / Standard Output)

“Westward Journey” is one of the most popular online games in China, which is developed and run by NIE. The game is based on the famous and romantic Chinese classical fiction “Journey to the West”, and the well-know film by Stephen Chow. The story behind “Westward Journey” is fantasy and thus attracts many players.

The game contains many regions, and different regions are ruled by different dominators. One of the regions which named “Tree World” is now ruled by a monster. There are N castles in this region, and each of them has its importance value (a positive integer not larger than 10^8). The castles are connected by (N – 1) bidirectional roads. The roads make all the castles connected (that means you can travel between any two castles of them). The importance values of the castles are variable. Now, the monster wants to know something if he destroys one of the roads. In detail, you are to handle totally Q instructions: each of them can be one of the following forms.

CHANGE i w The importance value of the

	Output **min_1** Explanations: The **jth** road can divide the “Tree World” into two connected components, named
QUERY j	**min_1**: the minimum importance value in **max_1:** the maximum importance value in **min_2**: the minimum importance value in **max_2:** the maximum importance value in

Input:

The first line contains an integer T ($T \le 10$), the number of test cases.

For each test case, the first line contains two integers N ($2 \le N \le 100000$) and Q ($1 \le Q \le 100000$), indicating the number of castles and instructions.

The following line contains N integers (positive integer not larger than 10^8), indicating the initial importance value of each castle (castles are numbered from 1 to N).

The following (N – 1) lines each contains two integers u and v, indicating castle u and castle v are connected directly by a bidirectional road (the roads are numbered from 1 to N - 1).

The following Q lines each contains an instruction according to the specification above.

Output:

For each “QUERY” instruction, output the result on a separate line.

Sample Input:

```
1
5 3
1 2 3 4 5
1 2
2 3
3 4
4 5
QUERY 1
CHANGE 1 10
QUERY 1
```

Sample Output:

```
11
110
```

Problem I
Counting
(Input File: i.in / Standard Output)

Recently, Anthony learnt counting. He took a paper and wrote down the sequential numbers from 1 to 10^n-1. For example, if n=2, he wrote:

"123456789101112131415...9899"

In this string, digit 1 appears 20 times, while digit 0 appears 9 times. To calculate the number of occurrences of a digit is an easy task for Anthony.

He is studying in a more complicate task: to calculate the number of occurrences of the number m which may be larger than 9. For example, if n=2 and m=78, 78 appears 3 times at "12345678...777879...868788...99".

Input:

Input contains several cases(less than 2000 cases).

Each case contains 2 integers n and m($1 \le n \le 15$, $1 \le m < 1000000$).

Input is terminated by two single zeros: “0 0”.

Output:

Output one integer for each case, representing the total number.

Sample input:

```
2 1
2 10
2 11
2 78
0 0
```

Sample output:

```
20
1
2
3
```

Problem J

Uncle Hey

(Input File: j.in / Standard Output)

The annual GaoDaoKuOI (GDKOI), an interesting programming contest, was held several weeks ago. Uncle Hey had been supposed to be a judge, but turned out to be a scheduler, which annoyed him very much. What's going on?

During the contest, contestants might want to go out for a break. (Don't ask why they have such a privilege. They do have anyway) However, it is not allowed that two or more contestants went out together. So, they had to go one by one, that is, First Come First Served. In case that two or more contestants posted their requests at the same time, you could safely assume that the one with the lexicographically smaller name came earlier.

It's an annoying job, isn't it? So, Uncle Hey wants to share the annoyance with you guys here.

Given the record of all requests like this:

T1 C1 D1

T2 C2 D2

…

Tn Cn Dn

Each line describes a request, where Ti represents the posting time of the request, and Ci represents the name of the contestant who posted the request, and Di means how long will it take for he/she to go out and come back. Note that, requests are given in no order.

You're supposed to report all events like this: "xxx went out at time yyy" (quotes for clarity), where xxx represents the name of the contestant and yyy is the time when he/she was permitted to go. You should print the events in the non-decreasing order of the happening time. Please refer to the sample for more information.

Input:

First line contains an integer M ($M \leq 6$), indicating the number of test cases.

Each test case is in the format like this:

First comes a positive integer N ($N \leq 10000$), indicating the number of requests.Then N lines follows. Each line contains a positive integer T, a string C and a positive integer D, representing a request. $T \leq 10000000$, $D \leq 1000$, C contains no more than 20 lowercase letters.

It's guaranteed that there are no two requests share a same name.

Output:

For each test case, print all events in the order described as above, one event per line. **Print a blank line after each test case.**

Sample Input:

```
2
1
10 lq 5
```

2
10 lq 5
5 wing 6

Sample Output:

lq went out at time 10

wing went out at time 5
lq went out at time 11