Problem Set: Problem Set for JAG Practice Contest 2009

Japanese Alumni Group

Contest Held on 01 Nov 2009

Problem A
Whist
Input: A.txt

Whist is a game played by four players with a standard deck of playing cards. The players seat around a table, namely, in north, east, south, and west. This game is played in a team-play basis: the players seating opposite to each other become a team. In other words, they make two teams we could call the north-south team and the east-west team.

Remember that the standard deck consists of 52 cards each of which has a rank and a suit. The rank indicates the strength of the card and is one of the following: 2, 3, 4, 5, 6, 7, 8, 9, 10, jack, queen, king, and ace (from the lowest to the highest). The suit refers to the type of symbols printed on the card, namely, spades, hearts, diamonds, and clubs. The deck contains exactly one card for every possible pair of a rank and a suit, thus 52 cards.

One of the four players (called a dealer) shuffles the deck and deals out all the cards face down, one by one, clockwise from the player left to him or her. Each player should have thirteen cards. Then the last dealt card, which belongs to the dealer, is turned face up. The suit of this card is called trumps and has a special meaning as mentioned below.

A deal of this game consists of thirteen tricks. The objective for each team is winning more tricks than another team. The player left to the dealer leads the first trick by playing one of the cards in his or her hand. Then the other players make their plays in the clockwise order. They have to play a card of the suit led if they have one; they can play any card otherwise. The trick is won by the player with the highest card of the suit led if no one plays a trump, or with the highest trump otherwise. The winner of this trick leads the next trick, and the remaining part of the deal is played similarly. After the thirteen tricks have been played, the team winning more tricks gains a score, one point per trick in excess of six.

Your task is to write a program that determines the winning team and their score for given plays of a deal.

Input

The input is a sequence of datasets. Each dataset corresponds to a single deal and has the following format:

> $Trump$
> $Card_{\mathrm{N},1}$ $Card_{\mathrm{N},2}$ $\ldots$ $Card_{\mathrm{N},13}$
> $Card_{\mathrm{E},1}$ $Card_{\mathrm{E},2}$ $\ldots$ $Card_{\mathrm{E},13}$

$Card_{S,1}$ $Card_{S,2}$ ... $Card_{S,13}$
$Card_{W,1}$ $Card_{W,2}$ ... $Card_{W,13}$

Trump indicates the trump suit. $Card_{N,i}$, $Card_{E,i}$, $Card_{S,i}$, and $Card_{W,i}$ denote the card played in the i-th trick by the north, east, south, and west players respectively. Each card is represented by two characters; the first and second character indicates the rank and the suit respectively.

The rank is represented by one of the following characters: '2', '3', '4', '5', '6', '7', '8', '9', 'T' (10), 'J' (jack), 'Q' (queen), 'K' (king), and 'A' (ace). The suit is represented by one of the following characters: 'S' (spades), 'H' (hearts), 'D' (diamonds), and 'C' (clubs).

You should assume the cards have been dealt out by the west player. Thus the first trick is led by the north player. Also, the input does not contain any illegal plays.

The input is terminated by a line with "#". This is not part of any dataset and thus should not be processed.

Output

For each dataset, print the winner and their score of the deal in a line, with a single space between them as a separator. The winner should be either "NS" (the north-south team) or "EW" (the east-west team). No extra character or whitespace should appear in the output.

Sample Input

```
H
4C 8H QS 5D JD KS 8S AH 6H 7H 3S 7S 6D
TC JC JS KD AC QC QD 2H QH 3H 3C 7C 4D
6C 9C AS TD 5H 6S 5S KH TH AD 9S 8D 2D
8C 5C 2S 7D KC 4S TS JH 4H 9H 2C 9D 3D
D
8D 9D 9S QS 4H 5H JD JS 9H 6S TH 6H QH
QD 9C 5S 7S 7H AC 2D KD 6C 3D 8C TC 7C
5D QC 3S 4S 3H 3C 6D KS JC AS 5C 8H TS
4D 4C 8S 2S 2H KC TD JH 2C AH 7D AD KH
#
```

Output for the Sample Input

```
EW 1
EW 2
```

The winners of the tricks in the first dataset are as follows (in the order of tricks): east, north, south, east, south, north, west, north, east, west, east, east, north.

Problem B
For the Peace
Input: B.txt

This is a story of a world somewhere far from the earth. In this world, the land is parted into a number of countries ruled by empires. This world is not very peaceful: they have been involved in army race.

They are competing in production of missiles in particular. Nevertheless, no countries have started wars for years. Actually they have a reason they can't get into wars — they have missiles much more than enough to destroy the entire world. Once a war would begin among countries, none of them could remain.

These missiles have given nothing but scare to people. The competition has caused big financial and psychological pressure to countries. People have been tired. Military have been tired. Even empires have been tired. No one wishes to keep on missile production.

So empires and diplomats of all countries held meetings quite a few times toward renouncement of missiles and abandon of further production. The meetings were quite difficult as they have different matters. However, they overcame such difficulties and finally came to the agreement of a treaty. The points include:

- Each country will dispose all the missiles of their possession by a certain date.
- The war potential should not be different by greater than a certain amount d among all countries.

Let us describe more details on the second point. Each missile has its capability, which represents how much it can destroy the target. The war potential of each country is measured simply by the sum of capability over missiles possessed by that country. The treaty requires the difference to be not greater than d between the maximum and minimum potential of all the countries.

Unfortunately, it is not clear whether this treaty is feasible. Every country is going to dispose their missiles only in the order of time they were produced, from the oldest to the newest. Some missiles have huge capability, and disposal of them may cause unbalance in potential.

Your task is to write a program to see this feasibility.

Input

The input is a sequence of datasets. Each dataset is given in the following format:

$n\ d$
$m_1\ c_{1,1}\ \dots\ c_{1,m_1}$
$\dots$
$m_n\ c_{n,1}\ \dots\ c_{n,m_n}$

The first line contains two positive integers n and d, the number of countries and the tolerated difference of potential ($n \leq 100$, $d \leq 1000$). Then n lines follow. The i-th line begins with a non-negative integer m_i, the number of the missiles possessed by the i-th country. It is followed by a sequence of m_i positive integers. The j-th integer $c_{i,j}$ represents the capability of the j-th *newest* missile of the i-th country ($c_{i,j} \leq 1000$). These integers are separated by a single space. Note that the country disposes their missiles in the reverse order of the given sequence.

The number of missiles is not greater than 10000. Also, you may assume the difference between the maximum and minimum potential does not exceed d in any dataset.

The input is terminated by a line with two zeros. This line should not be processed.

Output

For each dataset, print a single line. If they can dispose all their missiles according to the treaty, print “`Yes`” (without quotes). Otherwise, print “`No`”.

Note that the judge is performed in a *case-sensitive* manner. No extra space or character is allowed.

Sample Input

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

Output for the Sample Input

```
Yes
No
```

Problem C
Champernowne Constant
Input: C.txt

Champernown constant is an irrational number represented in decimal by "0." followed by concatenation of all positive integers in the increasing order. The first few digits of this constant are: 0.123456789101112 . . .

Your task is to write a program that outputs the K digits of Chapnernown constant starting at the N-th place for given two natural numbers K and N.

Input

The input has multiple lines. Each line has two positive integers N and K ($N \leq 10^9$, $K \leq 100$) separated by a space.

The end of input is indicated by a line with two zeros. This line should not be processed.

Output

For each line, output a line that contains the K digits.

Sample Input

```
4 5
6 7
0 0
```

Output for the Sample Input

```
45678
6789101
```

Problem D
Futon
Input: D.txt

The sales department of Japanese Ancient Giant Corp. is visiting a hot spring resort for their recreational trip. For deepening their friendships, they are staying in one large room of a Japanese-style hotel called a ryokan.

In the ryokan, people sleep in Japanese-style beds called futons. They all have put their futons on the floor just as they like. Now they are ready for sleeping but they have one concern: they don't like to go into their futons with their legs toward heads — this is regarded as a bad custom in Japanese tradition. However, it is not obvious whether they can follow a good custom. You are requested to write a program answering their question, as a talented programmer.

Here let's model the situation. The room is considered to be a grid on an xy-plane. As usual, x-axis points toward right and y-axis points toward up. Each futon occupies two adjacent cells. People put their pillows on either of the two cells. Their heads come to the pillows; their foots come to the other cells. If the cell of some person's foot becomes adjacent to the cell of another person's head, regardless their directions, then it is considered as a bad case. Otherwise people are all right.

Input

The input is a sequence of datasets. Each dataset is given in the following format:

n
x_1 y_1 dir_1
$\dots$
x_n y_n dir_n

n is the number of futons ($1 \leq n \leq 20,000$); (x_i, y_i) denotes the coordinates of the left-bottom corner of the i-th futon; dir_i is either '`x`' or '`y`' and denotes the direction of the i-th futon, where '`x`' means the futon is put horizontally and '`y`' means vertically. All coordinate values are non-negative integers not greater than 10^9.

It is guaranteed that no two futons in the input overlap each other.

The input is terminated by a line with a single zero. This is not part of any dataset and thus should not be processed.

Output

For each dataset, print “`Yes`” in a line if it is possible to avoid a bad case, or “`No`” otherwise.

Sample Input

```
4
0 0 x
2 0 x
0 1 x
2 1 x
4
1 0 x
0 1 x
2 1 x
1 2 x
4
0 0 x
2 0 y
0 1 y
1 2 x
0
```

Output for the Sample Input

```
Yes
No
Yes
```

Problem E
Safe Area
Input: E.txt

Nathan O. Davis is challenging a kind of shooter game. In this game, enemies emit laser beams from outside of the screen. A laser beam is a straight line with a certain thickness. Nathan moves a circular-shaped machine within the screen, in such a way it does not overlap a laser beam. As in many shooters, the machine is destroyed when the overlap happens.

Nathan is facing an uphill stage. Many enemies attack simultaneously in this stage, so eventually laser beams fill out almost all of the screen. Surprisingly, it is even possible he has no "safe area" on the screen. In other words, the machine cannot survive wherever it is located in some cases.

The world is as kind as it is cruel! There is a special item that helps the machine to survive any dangerous situation, even if it is exposed in a shower of laser beams, for some seconds. In addition, another straight line (called "a warning line") is drawn on the screen for a few seconds before a laser beam is emit along that line.

The only problem is that Nathan has a little slow reflexes. He often messes up the timing to use the special item. But he knows a good person who can write a program to make up his slow reflexes — it's you! So he asked you for help.

Your task is to write a program to make judgement whether he should use the item or not, for given warning lines and the radius of the machine.

Input

The input is a sequence of datasets. Each dataset corresponds to one situation with warning lines in the following format:

$W\ H\ N\ R$
$x_{1,1}\ y_{1,1}\ x_{1,2}\ y_{1,2}\ t_1$
$x_{2,1}\ y_{2,1}\ x_{2,2}\ y_{2,2}\ t_2$
...
$x_{N,1}\ y_{N,1}\ x_{N,2}\ y_{N,2}\ t_N$

The first line of a dataset contains four integers W, H, N and R ($2 < W \leq 640$, $2 < H \leq 480$, $0 \leq N \leq 100$ and $0 < R < \min\{W, H\}/2$). The first two integers W and H indicate the width and height of the screen, respectively. The next integer N represents the number of laser beams.

The last integer R indicates the radius of the machine. It is guaranteed that the output would remain unchanged if the radius of the machine would become larger by 10^{-5} than R.

The following N lines describe the N warning lines. The $(i+1)$-th line of the dataset corresponds to the i-th warning line, which is represented as a straight line which passes through two given different coordinates $(x_{i,1}, y_{i,1})$ and $(x_{i,2}, y_{i,2})$. The last integer t_i indicates the thickness of the laser beam corresponding to the i-th warning line.

All given coordinates (x, y) on the screen are a pair of integers $(0 \leq x \leq W,\ 0 \leq y \leq H)$. Note that, however, the machine is allowed to be located at non-integer coordinates during the play.

The input is terminated by a line with four zeros. This line should not be processed.

Output

For each case, print “`Yes`” in a line if there is a safe area, or print “`No`” otherwise.

Sample Input

```
100 100 1 1
50 0 50 100 50
640 480 1 1
0 0 640 480 100
0 0 0 0
```

Output for the Sample Input

```
No
Yes
```

Problem F
Water Tank
Input: F.txt

You built an apartment. The apartment has a water tank with a capacity of L in order to store water for the residents. The tank works as a buffer between the water company and the residents.

It is required to keep the tank "not empty" at least during use of water. A pump is used to provide water into the tank. From the viewpoint of avoiding water shortage, a more powerful pump is better, of course. But such powerful pumps are expensive. That's the life.

You have a daily schedule table of water usage. It does not differ over days. The table is composed of some schedules. Each schedule is indicated by the starting time of usage, the ending time and the used volume per unit of time during the given time span.

All right, you can find the minimum required speed of providing water for days from the schedule table. You are to write a program to compute it.

You can assume the following conditions.

- A day consists of 86,400 units of time.
- No schedule starts before the time 0 (the beginning of the day).
- No schedule ends after the time 86,400 (the end of the day).
- No two schedules overlap.
- Water is not consumed without schedules.
- The tank is full of water when the tank starts its work.

Input

The input is a sequence of datasets. Each dataset corresponds to a schedule table in the following format:

N L
s_1 t_1 u_1
. . .
s_N t_N u_N

The first line of a dataset contains two integers N and L ($1 \leq N \leq 86400$, $1 \leq L \leq 10^6$), which represents the number of schedule in the table and the capacity of the tank, respectively.

The following N lines describe the N schedules. The $(i+1)$-th line of the dataset corresponds to the i-th schedule, which consists of three integers s_i, t_i and u_i. The first two integers s_i and t_i indicate the starting time and the ending time of the schedule. The last integer u_i ($1 \leq u_i \leq 10^6$) indicates the consumed volume per unit of time during the schedule. It is guaranteed that $0 \leq s_1 < t_1 \leq s_2 < t_2 \leq \cdots \leq s_n < t_n \leq 86400$.

The input is terminated by a line with two zeros. This line should not be processed.

Output

For each case, print the minimum required amount of water per unit of time provided by the pump in a line. The amount may be printed with an arbitrary number of digits after the decimal point, but should not contain an absolute error greater than 10^{-6}.

Sample Input

```
1 100
0 86400 1
1 100
43200 86400 1
0 0
```

Output for the Sample Input

```
1.000000
0.997685
```

Problem G
Neko's Treasure
Input: G.txt

Maki is a house cat. One day she fortunately came at a wonderful-looking dried fish. Since she felt not hungry on that day, she put it up in her bed. However there was a problem; a rat was living in her house, and he was watching for a chance to steal her food. To secure the fish during the time she is asleep, she decided to build some walls to prevent the rat from reaching her bed.

Maki's house is represented as a two-dimensional plane. She has hidden the dried fish at (x_t, y_t). She knows that the lair of the rat is located at (x_s, y_s). She has some candidate locations to build walls. The i-th candidate is described by a circle of radius r_i centered at (x_i, y_i). She can build walls at as many candidate locations as she wants, unless they touch or cross each other. You can assume that the size of the fish, the rat's lair, and the thickness of walls are all very small and can be ignored.

Your task is to write a program which determines the minimum number of walls the rat needs to climb over until he can get to Maki's bed from his lair, assuming that Maki made an optimal choice of walls.

Input

The input is a sequence of datasets. Each dataset corresponds to a single situation and has the following format:

n
x_s y_s x_t y_t
x_1 y_1 r_1
$\dots$
x_n y_n r_n

n is the number of candidate locations where to build walls ($1 \leq n \leq 1000$). (x_s, y_s) and (x_t, y_t) denote the coordinates of the rat's lair and Maki's bed, respectively. The i-th candidate location is a circle which has radius r_i ($1 \leq r_i \leq 10000$) and is centered at (x_i, y_i) ($i = 1, 2, \dots, n$). All coordinate values are integers between 0 and 10000 (inclusive).

All candidate locations are distinct and contain neither the rat's lair nor Maki's bed. The positions of the rat's lair and Maki's bed are also distinct.

The input is terminated by a line with "`0`". This is not part of any dataset and thus should not be processed.

Output

For each dataset, print a single line that contains the minimum number of walls the rat needs to climb over.

Sample Input

```
3
0 0 100 100
60 100 50
100 100 10
80 80 50
4
0 0 100 100
50 50 50
150 50 50
50 150 50
150 150 50
0
```

Output for the Sample Input

```
2
0
```

Problem H
Eleven Lover
Input: H.txt

Edward Leven loves multiples of eleven very much. When he sees a number, he always tries to find consecutive subsequences (or substrings) forming multiples of eleven. He calls such subsequences as 11-sequences. For example, he can find an 11-sequence 781 in a number 17819.

He thinks a number which has many 11-sequences is a *good* number. He would like to find out a very *good* number. As the first step, he wants an easy way to count how many 11-sequences are there in a given number. Even for him, counting them from a big number is not easy. Fortunately, one of his friends, you, is a brilliant programmer. He asks you to write a program to count the number of 11-sequences. Note that an 11-sequence must be a positive number without leading zeros.

Input

The input is a sequence of lines each of which contains a number consisting of less than or equal to 80000 digits.

The end of the input is indicated by a line containing a single zero, which should not be processed.

Output

For each input number, output a line containing the number of 11-sequences.

You can assume the answer fits in a 32-bit signed integer.

Sample Input

```
17819
1111
11011
1234567891011121314151617181920
0
```

Output for the Sample Input

```
1
4
```

```
4
38
```

Problem I
Crystal Jails
Input: I.txt

Artistic Crystal Manufacture developed products named Crystal Jails. They are cool ornaments forming a rectangular solid. They consist of colorful crystal cubes. There are bright cores on the center of cubes, which are the origin of the name. The combination of various colored reflections shows fantastic dance of lights.

The company wanted to make big sales with Crystal Jails. They thought nice-looking color patterns were the most important factor for attractive products. If they could provide several nice patterns, some customers would buy more than one products. However, they didn't have staff who could design nice patterns. So they hired a temporary designer to decide the patterns.

The color pattern for mass production had a technical limitation: all cubes of the same color must be connected. In addition, they needed to send the pattern to the factory as a set of *blocks*, i.e. shapes formed by cubes of the same color. They requested him to represent the design in this form.

After a week of work, he sent various ideas of the color patterns to them. At first, his designs looked nice, but they noticed some patterns couldn't form a rectangular solid. He was a good designer, but they had not noticed he lacked space geometrical sense.

They didn't have time to ask him to revise his design. Although it was acceptable to ignore bad patterns, it also took a lot of time to figure out all bad patterns manually. So the leader of this project decided to ask you, a freelance programmer, for help.

Your task is to write a program to judge whether a pattern can form a rectangular solid. Note that blocks can be rotated.

Input

The input consists of multiple datasets. Each dataset is formatted as follows:

$W\ D\ H\ N$
$Block_1$
$Block_2$
...
$Block_N$

The first line of a dataset contains four positive integers W, D, H and N. W, D and H indicate the width, depth and height of a Crystal Jail. N indicates the number of colors.

The remaining lines describe N colored blocks. Each description is formatted as follows:

$$
\begin{array}{l}
w\ d\ h \\
c_{111}\ c_{211}\ \cdots\ c_{w11} \\
c_{121}\ c_{221}\ \cdots\ c_{w21} \\
\cdots \\
c_{1d1}\ c_{2d1}\ \cdots\ c_{wd1} \\
\\
c_{112}\ c_{212}\ \cdots\ c_{w12} \\
c_{122}\ c_{222}\ \cdots\ c_{w22} \\
\cdots \\
c_{1d2}\ c_{2d2}\ \cdots\ c_{wd2} \\
\\
\vdots \\
\\
c_{11h}\ c_{21h}\ \cdots\ c_{w1h} \\
c_{12h}\ c_{22h}\ \cdots\ c_{w2h} \\
\cdots \\
c_{1dh}\ c_{2dh}\ \cdots\ c_{wdh}
\end{array}
$$

The first line of the description contains three positive integers w, d and h, which indicate the width, depth and height of the block. The following $(d + 1) \times h$ lines describe the shape of the block. They show the cross-section layout of crystal cubes from bottom to top. On each height, the layout is described as d lines of w characters. Each character c_{xyz} is either '`*`' or '`.`'. '`*`' indicates there is a crystal cube on that space, and '`.`' indicates there is not. A blank line follows after each $(d \times w)$-matrix.

The input is terminated by a line containing four zeros.

You can assume the followings.

- $1 \leq W, D, H, w, d, h \leq 3$.
- $1 \leq N \leq 27$.
- Each block has at least one crystal cubes and they are connected.
- The total number of crystal cubes equals to $W \times D \times H$.

Output

For each dataset, determine whether the given blocks can form a $W \times D \times H$ rectangular solid and output “`Yes`” or “`No`” in one line.

Sample Input

```
3 3 3 5
3 2 2
***
.*.

.*.
...

3 2 1
***
**.

3 1 3
..*

.**

**.

3 2 2
..*
...

***
..*

3 1 3
.**

.**

***

3 3 3 2
3 3 3
***
***
***
```

```
***
*.*
***

***
***
***

1 1 1
*

3 2 1 2
3 1 1
***

2 2 1
**
*.

0 0 0 0
```

Output for the Sample Input

```
Yes
Yes
No
```

Problem J
Cave Explorer
Input: J.txt

Mike Smith is a man exploring caves all over the world.

One day, he faced a scaring creature blocking his way. He got scared, but in short time he took his knife and then slashed it to attempt to kill it. Then they were split into parts, which soon died out but the largest one. He slashed the creature a couple of times more to make it small enough, and finally became able to go forward.

Now let us think of his situation in a mathematical way. The creature is considered to be a polygon, convex or concave. Mike slashes this creature straight with his knife in some direction. We suppose here the direction is given for simpler settings, while the position is arbitrary. Then all split parts but the largest one disappears.

Your task is to write a program that calculates the area of the remaining part when the creature is slashed in such a way that the area is minimized.

Input

The input is a sequence of datasets. Each dataset is given in the following format:

n
v_x v_y
x_1 y_1
$\ldots$
x_n y_n

The first line contains an integer n, the number of vertices of the polygon that represents the shape of the creature ($3 \leq n \leq 100$). The next line contains two integers v_x and v_y, where (v_x, v_y) denote a vector that represents the direction of the knife ($-10000 \leq v_x, v_y \leq 10000$, $v_x^2 + v_y^2 > 0$). Then n lines follow. The i-th line contains two integers x_i and y_i, where (x_i, y_i) denote the coordinates of the i-th vertex of the polygon ($0 \leq x_i, y_i \leq 10000$).

The vertices are given in the counterclockwise order. You may assume the polygon is always simple, that is, the edges do not touch or cross each other except for end points.

The input is terminated by a line with a zero. This should not be processed.

Output

For each dataset, print the minimum possible area in a line. The area may be printed with an arbitrary number of digits after the decimal point, but should not contain an absolute error greater than 10^{-2}.

Sample Input

```
5
0 1
0 0
5 0
1 1
5 2
0 2
7
9999 9998
0 0
2 0
3 1
1 1
10000 9999
2 2
0 2
0
```

Output for the Sample Input

```
2.00
2.2500000
```