Problem A. North North West

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

We can describe detailed direction by repeating the directional names: north, south, east and west. For example, northwest is the direction halfway between north and west, and northnorthwest is between north and northwest.

In this problem, we describe more detailed direction between north and west as follows.

- "`north`" means 0 degrees.
- "`west`" means 90 degrees.
- If the direction dir means a degrees and the sum of the occurrences of "`north`" and "`west`" in dir is n (≥ 1), "`north`"dir (the concatenation of "`north`" and dir) means $a - \frac{90}{2^n}$ degrees and "`west`"dir means $a + \frac{90}{2^n}$ degrees.

Your task is to calculate the angle in degrees described by the given direction.

Input

The input contains several datasets. The number of datasets does not exceed 100.

Each dataset is described by a single line that contains a string denoting a direction. You may assume the given string can be obtained by concatenating some "`north`" and "`west`", the sum of the occurrences of "`north`" and "`west`" in the given string is between 1 and 20, inclusive, and the angle denoted by the given direction is between 0 and 90, inclusive. The final dataset is followed by a single line containing only a single '`#`'.

Output

For each dataset, print an integer if the angle described by the given direction can be represented as an integer, otherwise print it as an irreducible fraction. Follow the format of the sample output.

Examples

standard input	standard output

`north`
`west`
`northwest`
`northnorthwest`
`westwestwestnorth`
`#`
	`0`
	`90`
	`45`
	`45/2`
	`315/4`

Problem B. Unknown Switches

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 mebibytes

In the headquarter building of ICPC (International Company of Plugs & Connectors), there are M light bulbs and they are controlled by N switches. Each light bulb can be turned on or off by exactly one switch. Each switch may control multiple light bulbs. When you operate a switch, all the light bulbs controlled by the switch change their states. You lost the table that recorded the correspondence between the switches and the light bulbs, and want to restore it.

You decided to restore the correspondence by the following procedure.

- At first, every switch is off and every light bulb is off.
- You operate some switches represented by S_1.
- You check the states of the light bulbs represented by B_1.
- You operate some switches represented by S_2.
- You check the states of the light bulbs represented by B_2.
- ...
- You operate some switches represented by S_Q.
- You check the states of the light bulbs represented by B_Q.

After you operate some switches and check the states of the light bulbs, the states of the switches and the light bulbs are kept for next operations.

Can you restore the correspondence between the switches and the light bulbs using the information about the switches you have operated and the states of the light bulbs you have checked?

Input

The input consists of multiple datasets. The number of dataset is no more than 50 and the file size is no more than 10MB.

The first line of each dataset contains three integers N ($1 \le N \le 36$), M ($1 \le M \le 1{,}000$), Q ($0 \le Q \le 1{,}000$), which denote the number of switches, the number of light bulbs and the number of operations respectively. The following Q lines describe the information about the switches you have operated and the states of the light bulbs you have checked. The i-th of them contains two strings S_i and B_i of lengths N and M respectively. Each S_i denotes the set of the switches you have operated: S_{ij} is either 0 or 1, which denotes the j-th switch is not operated or operated respectively. Each B_i denotes the states of the light bulbs: B_{ij} is either 0 or 1, which denotes the j-th light bulb is off or on respectively.

You can assume that there exists a correspondence between the switches and the light bulbs which is consistent with the given information.

The end of input is indicated by a line containing three zeros.

Output

For each dataset, output the correspondence between the switches and the light bulbs consisting of M numbers written in base-36. In the base-36 system for this problem, the values 0-9 and 10-35 are represented by the characters '`0`'-'`9`' and '`A`'-'`Z`' respectively. The i-th character of the correspondence means the number of the switch controlling the i-th light bulb. If you cannot determine which switch controls the i-th light bulb, output '`?`' as the i-th character instead of the number of a switch.

Example

standard input	standard output
3 10 3	2222221100
000 0000000000	??
110 0000001111	0
101 1111111100	1
2 2 0	0123456789A
1 1 0	
2 1 1	
01 1	
11 11 10	
10000000000 10000000000	
11000000000 01000000000	
01100000000 00100000000	
00110000000 00010000000	
00011000000 00001000000	
00001100000 00000100000	
00000110000 00000010000	
00000011000 00000001000	
00000001100 00000000100	
00000000110 00000000010	
0 0 0	

Problem C. Speedrun

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

Infinite Chronicle -Princess Castle- is a simple role-playing game. There are $n+1$ checkpoints, numbered 0 through n, and for each $i = 1, 2, \ldots, n$, there is a unique one-way road running from checkpoint $i-1$ to i. The game starts at checkpoint 0 and ends at checkpoint n. Evil monsters will appear on the roads and the hero will have battles against them. You can save your game progress at any checkpoint; if you lose a battle, you can restart the game from the checkpoint where you have saved for the last time. At the beginning of the game, the progress is automatically saved at checkpoint 0 with no time.

Rabbit Hanako is fond of this game and now interested in speedrunning. Although Hanako is an expert of the game, she cannot always win the battles because of random factors. For each i, she estimated the probability p_i to win all the battles along the road from checkpoint $i-1$ to i. Everytime she starts at checkpoint $i-1$, after exactly one miniutes, she will be at checkpoint i with probability p_i and where she saved for the last time with probability $1-p_i$.

What puzzles Hanako is that it also takes one minute (!) to save your progress at a checkpoint, so it might be a good idea to pass some checkpoints without saving in order to proceed quickly. The task is to compute the minimum possible expected time needed to complete the game.

Input

The input consists of multiple datasets. The number of datasets is no more than 50. Each dataset has two lines: the first line contains an integer n ($1 \leq n \leq 10^5$), representing the number of roads, and the second line contains n numbers $p_1, p_2, \ldots, p_n$ ($0 < p_i \leq 1$), representing the winning probabilities. Each p_i has exactly two digits after the decimal point. The end of input is denoted as a line containing only a single zero.

Output

For each dataset, display the minimum expected time in minutes with a relative error of at most 10^{-8} in a line.

Example

standard input	standard output

2
0.50 0.40
2
0.70 0.60
4
0.99 1.00 1.00 0.01
0
	5.5000000000
	4.0476190476
	104.0101010101

Problem D. Flowers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 mebibytes

We have planted N flower seeds, all of which come into different flowers. We want to make all the flowers come out together.

Each plant has a value called vitality, which is initially zero. Watering and spreading fertilizers cause changes on it, and the i-th plant will come into flower if its vitality is equal to or greater than th_i. Note that th_i may be negative because some flowers require no additional nutrition.

Watering effects on all the plants. Watering the plants with W liters of water changes the vitality of the i-th plant by $W \times vw_i$ for all i $(1 \leq i \leq n)$, and costs $W \times pw$ yen, where W need not be an integer. vw_i may be negative because some flowers hate water.

We have N kinds of fertilizers, and the i-th fertilizer effects only on the i-th plant. Spreading F_i kilograms of the i-th fertilizer changes the vitality of the i-th plant by $F_i \times vf_i$, and costs $F_i \times pf_i$ yen, where F_i need not be an integer as well. Each fertilizer is specially made for the corresponding plant, therefore vf_i is guaranteed to be positive.

Of course, we also want to minimize the cost. Formally, our purpose is described as "to minimize $W \times pw + \sum_{i=1}^{N}(F_i \times pf_i)$ under $W \times vw_i + F_i \times vf_i \geq th_i$, $W \geq 0$, and $F_i \geq 0$ for all i $(1 \leq i \leq N)$". Your task is to calculate the minimum cost.

Input

The input consists of multiple datasets. The number of datasets does not exceed 100, and the data size of the input does not exceed 20MB.

The first line of a dataset contains a single integer N, number of flower seeds. The second line of a dataset contains a single integer pw, cost of watering one liter. Each of the following N lines describes a flower. The i-th line contains four integers, vw_i, pf_i, vf_i, and th_i, separated by a space.

You can assume that $1 \leq N \leq 10^5$, $1 \leq pw \leq 100$, $-100 \leq vw_i \leq 100$, $1 \leq pf_i \leq 100$, $1 \leq vf_i \leq 100$, and $-100 \leq th_i \leq 100$.

The end of the input is indicated by a line containing a zero.

Output

For each dataset, output a line containing the minimum cost to make all the flowers come out. The output must have an absolute or relative error at most 10^{-4}.

Examples

standard input	standard output

3
10
4 3 4 10
5 4 5 20
6 5 6 30
3
7
-4 3 4 -10
5 4 5 20
6 5 6 30
3
1
-4 3 4 -10
-5 4 5 -20
6 5 6 30
3
10
-4 3 4 -10
-5 4 5 -20
-6 5 6 -30
0
	43.5
	36
	13.5
	0

Problem E. Square in Circles

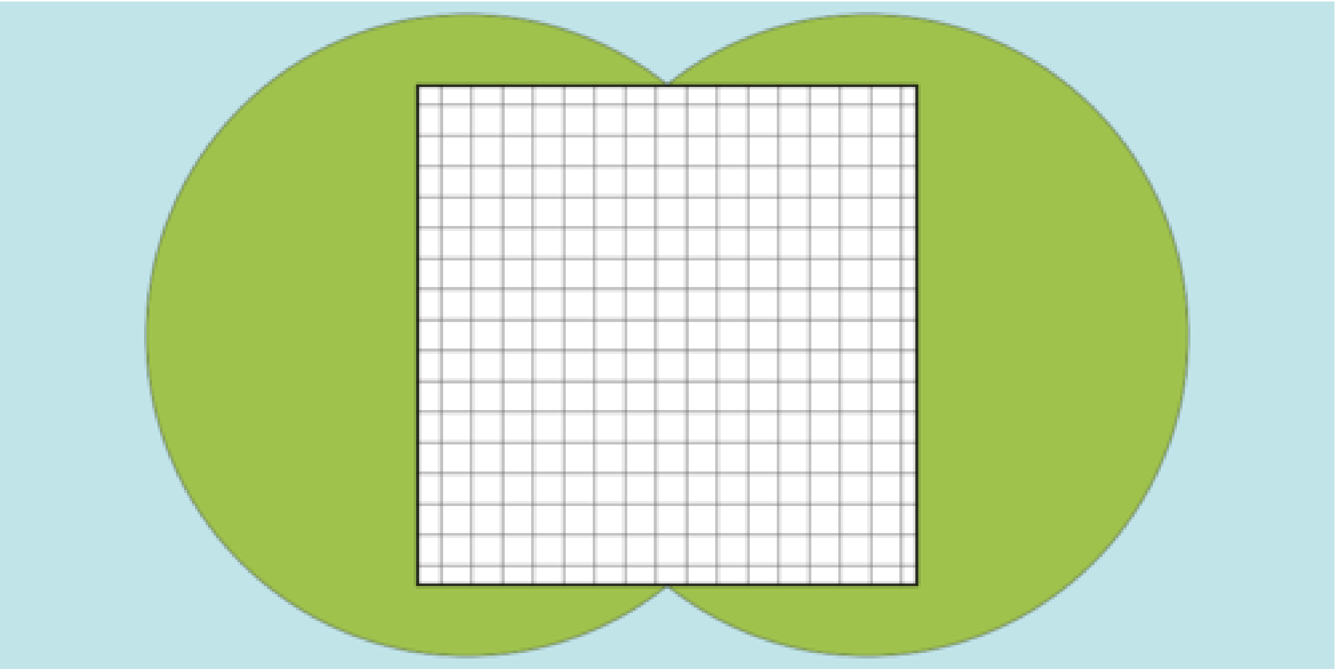
Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Eastern Circles Island is known for its mysterious shape: it is a completely flat island with its shape being a union of circles whose centers are on the x-axis and their inside regions.

The King of Eastern Circles Island plans to build a large square on Eastern Circles Island in order to celebrate the fiftieth anniversary of his accession. The King wants to make the square as large as possible. The whole area of the square must be on the surface of Eastern Circles Island, but any area of Eastern Circles Island can be used for the square. He also requires that the shape of the square is square (of course!) and at least one side of the square is parallel to the x-axis.

You, a minister of Eastern Circles Island, are now ordered to build the square. First, the King wants to know how large the square can be. You are given the positions and radii of the circles that constitute Eastern Circles Island. Answer the side length of the largest possible square.

N circles are given in an ascending order of their centers' x-coordinates. You can assume that for all i ($1 \leq i \leq N-1$), the i-th and $(i+1)$-st circles overlap each other. You can also assume that no circles are completely overlapped by other circles.



Picture above shows shape of Eastern Circles Island and one of the largest possible squares for test case 1 of sample input.

Input

The input consists of multiple datasets. The number of datasets does not exceed 30.

The first line of a dataset contains a single integer N ($1 \leq N \leq 50{,}000$), the number of circles that constitute Eastern Circles Island. Each of the following N lines describes a circle. The $(i+1)$-st line contains two integers X_i ($-100{,}000 \leq X_i \leq 100{,}000$) and R_i ($1 \leq R_i \leq 100{,}000$). X_i denotes the x-coordinate of the center of the i-th circle and R_i denotes the radius of the i-th circle. The y-coordinate of every circle is 0, that is, the center of the i-th circle is at $(X_i, 0)$.

You can assume the followings.

- For all i $(1 \leq i \leq N-1)$, X_i is strictly less than X_{i+1}.
- For all i $(1 \leq i \leq N-1)$, the i-th circle and the $(i+1)$-st circle have at least one common point $(X_{i+1} - X_i \leq R_i + R_{i+1})$.
- Every circle has at least one point that is not inside or on the boundary of any other circles.

The end of the input is indicated by a line containing a zero.

Output

For each dataset, output a line containing the side length of the square with the largest area. The output must have an absolute or relative error at most 10^{-4}.

Examples

standard input	standard output

2
0 8
10 8
2
0 7
10 7
0
	12.489995996796796
	9.899494936611665

Problem F. Reverse a Road II

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

JAG Kingdom is a strange kingdom such that its N cities are connected only by one-way roads. The N cities are numbered 1 through N. ICPC (International Characteristic Product Corporation) transports its products from the factory at the city S to the storehouse at the city T in JAG Kingdom every day. For efficiency, ICPC uses multiple trucks at once. Each truck starts from S and reaches T on the one-way road network, passing through some cities (or directly). In order to reduce risks of traffic jams and accidents, no pair of trucks takes the same road.

Now, ICPC wants to improve the efficiency of daily transports, while ICPC operates daily transports by as many trucks as possible under the above constraint. JAG Kingdom, whose finances are massively affected by ICPC, considers to change the direction of one-way roads in order to increase the number of trucks for daily transports of ICPC. Because reversal of many roads causes confusion, JAG Kingdom decides to reverse at most a single road.

If there is no road such that reversal of the road can improve the transport efficiency, JAG Kingdom need not reverse any roads. Check whether reversal of a single road can improve the current maximum number of trucks for daily transports. And if so, calculate the maximum number of trucks which take disjoint sets of roads when a one-way road can be reversed, and the number of roads which can be chosen as the road to be reversed to realize the maximum.

Input

The input consists of multiple datasets. The number of dataset is no more than 100.

The first line of each dataset contains four integers: the number of cities N ($2 \leq N \leq 1{,}000$), the number of roads M ($1 \leq M \leq 10{,}000$), the city with the factory S and the city with the storehouse T ($1 \leq S, T \leq N$, $S \neq T$).

The following M lines describe the information of the roads. The i-th line of them contains two integers a_i and b_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$), meaning that the i-th road is directed from a_i to b_i.

The end of input is indicated by a line containing four zeros.

Output

For each dataset, output two integers separated by a single space in a line as follows: If reversal of a single road improves the current maximum number of trucks for daily transports, the first output integer is the new maximum after reversal of a road, and the second output integer is the number of roads which can be chosen as the road to be reversed to realize the new maximum. Otherwise, i.e. if the current maximum cannot be increased by any reversal of a road, the first output integer is the current maximum and the second output integer is 0.

Examples

standard input	standard output

4 4 1 4
1 2
3 1
4 2
3 4
7 8 1 7
1 2
1 3
2 4
3 4
4 5
4 6
5 7
7 6
6 4 5 2
1 2
1 3
4 5
5 6
10 21 9 10
9 1
9 2
9 3
9 4
10 1
10 2
10 3
10 4
1 5
2 5
2 6
3 6
3 7
4 7
4 8
1 8
5 10
6 10
7 10
10 8
10 9
2 15 1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
1 2
0 0 0 0
	1 2
	2 1
	0 0
	4 6
	15 0

Problem G. Cookie Counter

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

One day, my grandmas left N cookies. My elder sister and I were going to eat them immediately, but there was the instruction. It said

- Cookies will go bad; you should eat all of them within D days.
- Be careful about overeating; you should eat strictly less than X cookies in a day.

My sister said "`How many ways are there to eat all of the cookies? Let's try counting!`"

Two ways are considered different if there exists a day such that the numbers of the cookies eaten on that day are different in the two ways. For example, if N, D and X are 5, 2 and 5 respectively, the number of the ways is 4:

- Eating 1 cookie on the first day and 4 cookies on the second day.
- Eating 2 cookies on the first day and 3 cookies on the second day.
- Eating 3 cookies on the first day and 2 cookies on the second day.
- Eating 4 cookies on the first day and 1 cookie on the second day.

I noticed the number of the ways would be very huge and my sister would die before counting it. Therefore, I tried to count it by a computer program to save the life of my sister.

Input

The input consists of multiple datasets. The number of datasets is no more than 100. For each dataset, three numbers N ($1 \le N \le 2{,}000$), D ($1 \le D \le 10^{12}$) and X ($1 \le X \le 2{,}000$) are written in a line and separated by a space. The end of input is denoted by a line that contains three zeros.

Output

Print the number of the ways modulo 1,000,000,007 in a line for each dataset.

Examples

standard input	standard output
5 2 5	4
3 3 3	7
5 4 5	52
4 1 2	0
1 5 1	0
1250 50 50	563144298
0 0 0	

Problem H. Points and Lines

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

One day, you found an old scroll with strange texts on it.

You revealed that the text was actually an expression denoting the position of the hidden treasure. The expression consists of following three operations:

- From two points, yield a line on which the points lie.
- From a point and a line, yield a point that is symmetric to the given point with respect to the line.
- From two lines, yield a point that is the intersection of the lines.

The syntax of the expression is denoted by following BNF:

```
<expression>      ::= <point>
<point> ::= <point-factor> | <line> "@" <line-factor> | <line> "@" <point-factor> | <point> "@" <line-factor>
<point-factor>    ::= "(" <number> "," <number> ")" | "(" <point> ")"
<line>            ::= <line-factor> | <point> "@" <point-factor>
<line-factor>     ::= "(" <line> ")"
<number>          ::= <zero-digit> | <positive-number> | <negative-number>
<positive-number> ::= <nonzero-digit> | <positive-number> <digit>
<negative-number> ::= "-" <positive-number>
<digit>           ::= <zero-digit> | <nonzero-digit>
<zero-digit>      ::= "0"
<nonzero-digit>   ::= "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

Each <point> or <point-factor> denotes a point, whereas each <line> or <line-factor> denotes a line. The former notion of <point-factor> (X, Y) represents a point which has X for x-coordinate and Y for y-coordinate on the 2-dimensional plane. "@" indicates the operations on two operands. Since each operation is distinguishable from others by its operands' types (i.e. a point or a line), all of these operations are denoted by the same character "@". Note that "@" is left-associative, as can be seen from the BNF.

Your task is to determine where the treasure is placed.

Input

The input consists of multiple datasets. Each dataset is a single line which contains an expression denoting the position of treasure.

It is guaranteed that each dataset satisfies the following conditions:

- The length of the string never exceeds 10^2.
- If both operands of "@" are points, their distance is greater than 1.
- If both operands of "@" are lines, they are never parallel.
- The absolute values of points' coordinates never exceed 10^2 at any point of evaluation.

You can also assume that there are at most 100 datasets.

The input ends with a line that contains only a single "#".

Output

For each dataset, print the X and Y coordinates of the point, denoted by the expression, in this order.

The output will be considered correct if its absolute or relative error is at most 10^{-2}.

Examples

standard input	standard output
`((0,0)@(1,1))@((4,1)@(2,5))`	`3.00000000 3.00000000`
`((0,0)@(3,1))@((1,-3)@(2,-1))`	`3.00000000 1.00000000`
`(0,0)@(1,1)@(4,1)`	`1.00000000 4.00000000`
`(0,0)@((1,1)@(4,1))`	`0.00000000 2.00000000`
`(((0,0)@((10,20)@(((30,40))))))`	`-10.00000000 10.00000000`
`((0,0)@(3,1))@((1,-3)@(2,-1))@(100,-100)@(100,100)`	`-99.83681795 -91.92248853`
`#`	

Problem I. Color the Map Extreme

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	512 mebibytes

You have just transferred to another world, and got a map of this world. There are several countries in this world. Each country has a connected territory, which is drawn on the map as a simple polygon consisting of its border segments in the 2-dimensional plane.

You are strange to this world, so you would like to paint countries on the map to distinguish them. If you paint adjacent countries with same color, it would be rather difficult to distinguish the countries. Therefore, you want to paint adjacent countries with different colors. Here, we define two countries are adjacent if their borders have at least one common segment whose length is strictly greater than 0. Note that two countries are NOT considered adjacent if the borders only touch at points.

Because you don't have the currency of this world, it is hard for you to prepare many colors. What is the minimum number of colors to paint the map such that adjacent countries can be painted with different colors?

Input

The input consists of multiple datasets. The number of dataset is no more than 35.

The first line of each dataset contains an integer n ($1 \le n \le 35$), which denotes the number of countries in this world.

The rest of each dataset describes the information of n polygons representing the countries. The first line of the information of the i-th polygon contains an integer m_i ($3 \le m_i \le 50$), which denotes the number of vertices. The following m_i lines describe the coordinates of the vertices in the counter-clockwise order. The j-th line of them contains two integers $x_{i,j}$ and $y_{i,j}$ ($|x_{i,j}|, |y_{i,j}| \le 10^3$), which denote the coordinates of the j-th vertex of the i-th polygon.

You can assume the followings.

- Each polygon has an area greater than 0.
- Two vertices of the same polygon have distinct coordinates.
- Two segments of the same polygon do not have any common points except that exactly two segments meet at each vertex.
- Two polygons have no common area.

The end of input is indicated by a line containing a single zero.

Output

For each dataset, output the minimum number of colors to paint the map such that adjacent countries are painted with different colors in a line.

Examples

standard input	standard output

1
3
0 0
1 0
0 1
4
4
0 0
10 0
10 10
0 10
4
10 0
20 0
20 10
10 10
4
0 10
10 10
10 20
0 20
4
10 10
20 10
20 20
10 20
3
4
-10 -10
2 2
10 10
-11 7
3
-1 -1
1 -1
0 0
3
0 0
3 -3
20 20
0
	1
	2
	3

standard input	standard output

7
4
46 12
52 12
53 15
45 15
32
67 1
70 0
73 1
77 3
79 5
80 8
77 8
76 5
74 4
71 3
70 3
67 4
65 6
63 8
62 14
64 19
66 21
70 22
75 21
78 16
80 16
80 17
79 20
78 22
74 24
67 24
63 22
61 19
60 15
60 10
62 5
64 3
5
74 14
80 14
80 16
78 16
74 16
19
34 0
37 0
37 19
36 22
35 23
32 24
30 24
27 24
25 23
23 20
23 18
23 15
26 15
26 18
27 20
29 21
32 21
34 20
34 18
4
47 0
50 0
42 24
39 24
4
79 20
80 17
80 22
78 22
4
50 0
58 24
56 24
49 3
0
	3

standard input	standard output

4
10
34 21
34 14
35 14
35 19
40 19
40 20
35 20
35 22
30 22
30 21
16
20 24
21 24
21 33
42 33
42 20
40 20
40 19
45 19
45 5
40 5
40 4
46 4
46 20
43 20
43 34
20 34
10
26 21
26 14
27 14
27 21
30 21
30 22
21 22
21 24
20 24
20 21
12
34 8
34 4
40 4
40 5
35 5
35 14
34 14
34 9
27 9
27 14
26 14
26 8
0
	4

In system all three samples are merged in one (with 5 integers in the answer).

Problem J. Website Tour

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 mebibytes

You want to compete in ICPC (Intergalactic Contest of Point Collection). In this contest, we move around in N websites, numbered 1 through N, within a time limit and collect points as many as possible. We can start and end on any website.

There are M links between the websites, and we can move between websites using these links. You can assume that it doesn't take time to move between websites. These links are directed and websites may have links to themselves.

In each website i, there is an advertisement and we can get p_i point(s) by watching this advertisement in t_i seconds. When we start on or move into a website, we can decide whether to watch the advertisement or not. But we cannot watch the same advertisement more than once before using any link in the website, while we can watch it again if we have moved among websites and returned to the website using one or more links, including ones connecting a website to itself. Also we cannot watch the advertisement in website i more than k_i times.

You want to win this contest by collecting as many points as you can. So you decided to compute the maximum points that you can collect within T seconds.

Input

The input consists of multiple datasets. The number of dataset is no more than 60.

The first line of each dataset contains three integers N ($1 \le N \le 100$), M ($0 \le M \le 1{,}000$) and T ($1 \le T \le 10{,}000$), which denote the number of websites, the number of links, and the time limit, respectively. All the time given in the input is expressed in seconds.

The following N lines describe the information of advertisements. The i-th of them contains three integers p_i ($1 \le p_i \le 10{,}000$), t_i ($1 \le t_i \le 10{,}000$) and k_i ($1 \le k_i \le 10{,}000$), which denote the points of the advertisement, the time required to watch the advertisement, and the maximum number of times you can watch the advertisement in website i, respectively.

The following M lines describe the information of links. Each line contains two integers a_i and b_i ($1 \le a_i, b_i \le N$), which mean that we can move from website a_i to website b_i using a link.

The end of input is indicated by a line containing three zeros.

Output

For each dataset, output the maximum points that you can collect within T seconds.

Examples

standard input	standard output

5 4 10
4 3 1
6 4 3
3 2 4
2 2 1
8 5 3
1 2
2 3
3 4
4 5
3 3 1000
1000 1 100
1 7 100
10 9 100
1 2
2 3
3 2
1 0 5
25 25 2
1 0 25
25 25 2
5 5 100
1 1 20
1 1 20
10 1 1
10 1 1
10 1 1
1 2
2 1
3 4
4 5
5 3
3 3 100
70 20 10
50 15 20
90 10 10
1 2
2 2
2 3
0 0 0
	15
	2014
	0
	25
	40
	390

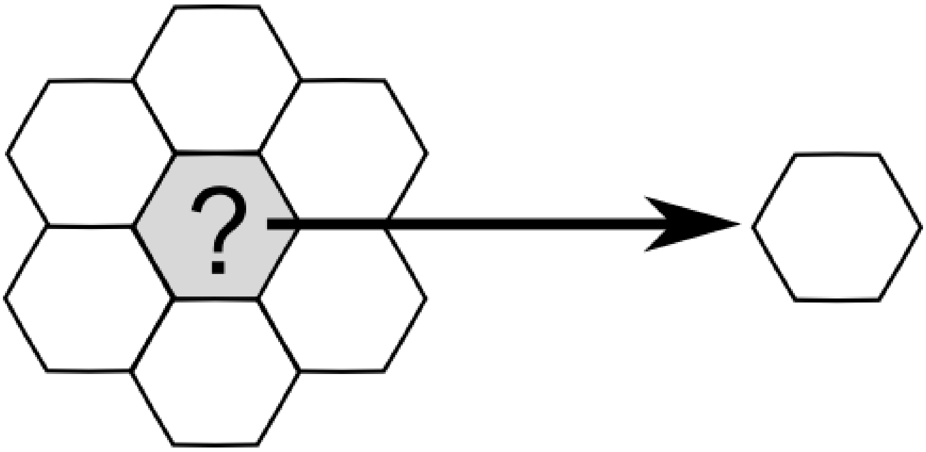
Problem K. Idempotent Filter

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

Let's consider operations on monochrome images that consist of hexagonal pixels, each of which is colored in either black or white. Because of the shape of pixels, each of them has exactly six neighbors (e.g. pixels that share an edge with it.)

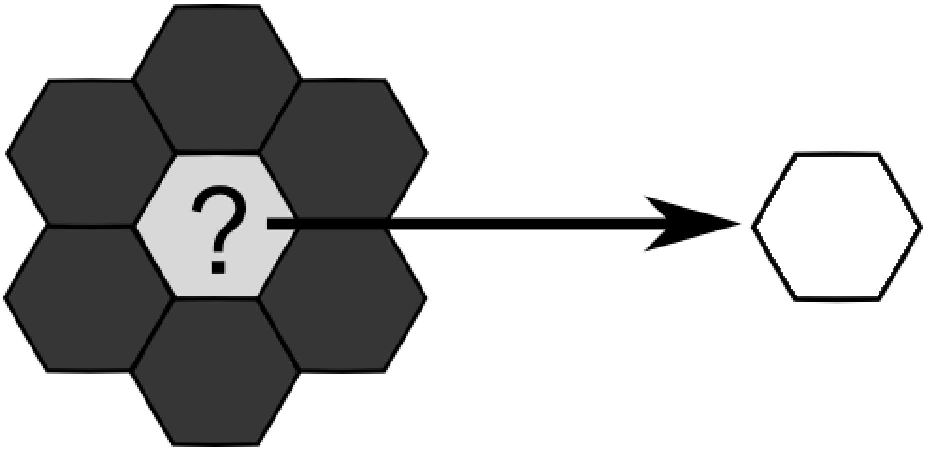
"`Filtering`" is an operation to determine the color of a pixel from the colors of itself and its six neighbors. Examples of filterings are shown below.

Example 1: Color a pixel in white when all of its neighboring pixels are white. Otherwise the color will not change.



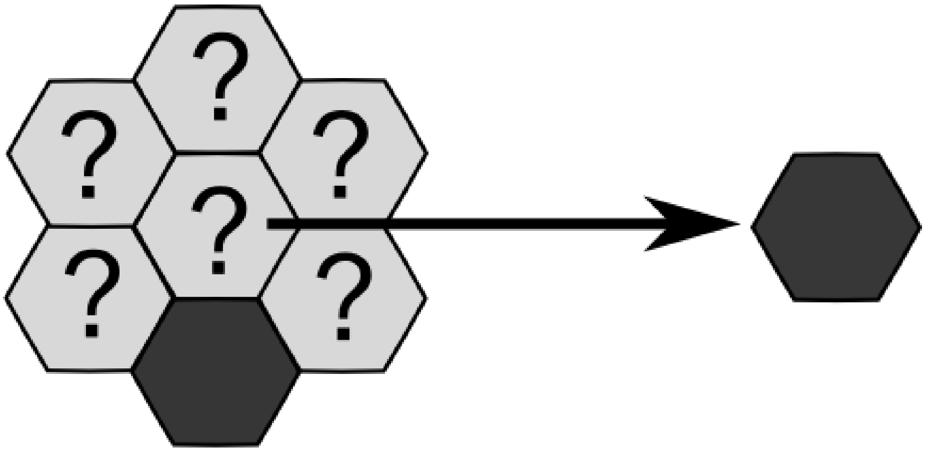
Performing this operation on all the pixels simultaneously results in "`noise canceling,`" which removes isolated black pixels.

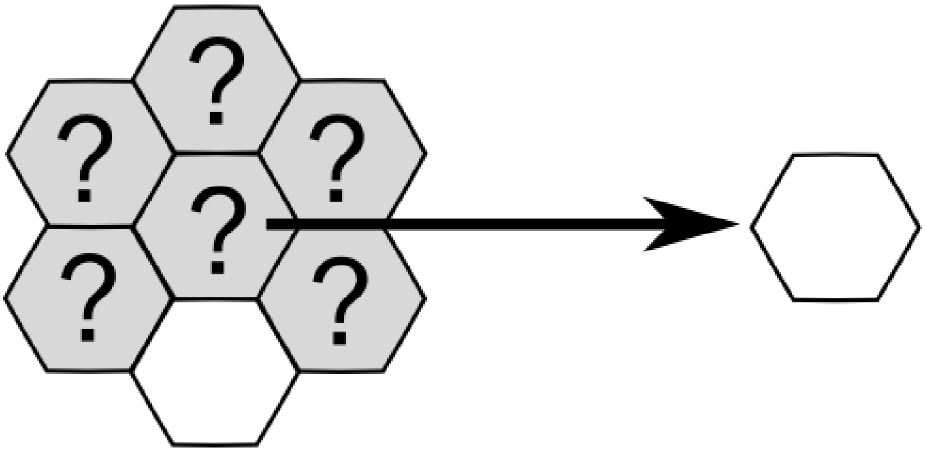
Example 2: Color a pixel in white when its all neighboring pixels are black. Otherwise the color will not change.



Performing this operation on all the pixels simultaneously results in "`edge detection,`" which leaves only the edges of filled areas.

Example 3: Color a pixel with the color of the pixel just below it, ignoring any other neighbors.





Performing this operation on all the pixels simultaneously results in “`shifting up`” the whole image by one pixel.

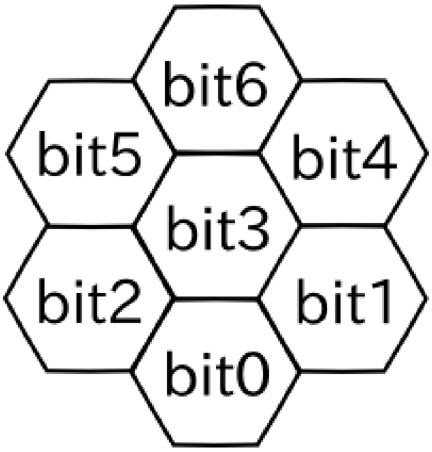
Applying some filter, such as “`noise canceling`” and “`edge detection,`” twice to any image yields the exactly same result as if they were applied only once. We call such filters *idempotent*. The “`shifting up`” filter is not idempotent since every repeated application shifts the image up by one pixel.

Your task is to determine whether the given filter is idempotent or not.

Input

The input consists of multiple datasets. The number of dataset is less than 100. Each dataset is a string representing a filter and has the following format (without spaces between digits).

c_i is either ‘`0`’ (represents black) or ‘`1`’ (represents white), which indicates the output of the filter for a pixel when the binary representation of the pixel and its neighboring six pixels is i. The mapping from the pixels to the bits is as following:



and the binary representation i is defined as $i = \sum_{j=0}^{6} bit_j \times 2^j$, where bit_j is 0 or 1 if the corresponding pixel is in black or white, respectively. Note that the filter is applied on the center pixel, denoted as bit 3.

The input ends with a line that contains only a single "#".

Output

For each dataset, print "yes" in a line if the given filter is idempotent, or "no" otherwise (quotes are for clarity).

Examples

standard input
00000000111111110000000011111111000000001111111100000000111111110000000011111111000000001111111100000000111111110000000011111111 10000000111111110000000011111111000000001111111100000000111111110000000011111111000000001111111100000000111111110000000011111111 01010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010101 #
standard output
yes yes no