



acm International Collegiate
Programming Contest

2005



event
sponsor

ACM International Collegiate Programming Contest 2005

South American Regional

November 11-12, 2005

Contest Session

(This problem set contains 9 problems; pages are numbered from 1 to 17)

Contest sites:

- Argentina
- Brazil
- Chile
- Colombia
- Peru
- Venezuela

Problem A

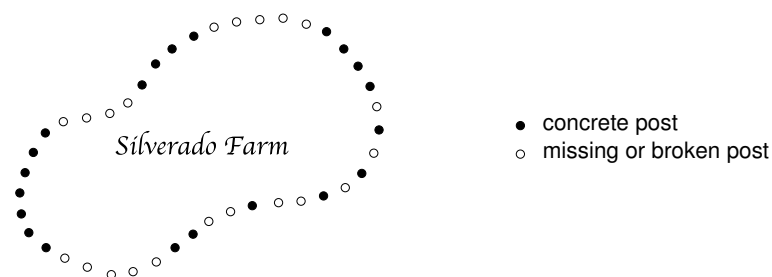
Tornado!

Source file name: `tornado.c`, `tornado.cpp`, `tornado.java` or `tornado.pas`

Is this crazy weather the result of mankind's continuous interference in the environment? Or is it simply the normal cycle of climate changes through the ages? No one seems to know for sure, but the fact is that natural phenomena such as tornadoes and hurricanes have been hitting our country with more force and frequency than in past decades.

One tornado has just hit Silverado Farm, a cattle and milk producer, and made havoc. The barn roof was torn, several trees were uprooted, the farm truck was overturned... But the worst thing is that the tornado destroyed several sections of the fence that surrounded the property. The fence was very well built, with concrete posts every two meters, and barbed wire enclosing the whole farm perimeter (the perimeter, in meters, is an even number, making the fence perfectly regular).

Now several posts are broken or missing, and there are gaps in the fence. To prevent the cattle from getting out of the property, the fence must be restored as quickly as possible. Reconstructing the fence to its original form, with concrete posts, will take a long time. In the meantime, the farm owners decided to close the gaps with a temporary fence, made with wooden posts. Wooden posts will be placed in exactly the same spots where missing/broken concrete posts were/are. However, in order to make the temporary reconstruction faster and less expensive, the owners decided to use fewer posts: a wooden post will be used to replace a missing/broken concrete post only if the length of the barbed wired needed to close the distance to the next post (wooden or concrete) exceeds four meters.



Given the description of which posts are missing/broken, you must write a program to determine the smallest number of wooden posts needed to close all the gaps in the fence, according to the owners' decision.

Input

The input contains several test cases. The first line of a test case contains one integer N indicating the number of original concrete posts in the fence ($5 \leq N \leq 5000$). The second line of a test case contains N integers X_i indicating the state of each concrete post after the tornado ($0 \leq X_i \leq 1$ for $1 \leq i \leq N$). If $X_i = 1$ post i is in good condition, if $X_i = 0$ post i is broken or missing. Note that post N is next to post 1. The end of input is indicated by $N = 0$.

The input must be read from standard input.

Output

For each test case in the input your program must produce one line of output, containing an integer indicating the smallest number of wooden posts that are needed to restore the fence, according to the owners' decision.

The output must be written to standard output.

Sample input	Output for the sample input
10	2
1 0 0 1 0 0 1 0 1 1	2
11	3
1 0 0 1 0 0 0 1 1 0 1	
12	
0 0 0 0 0 1 1 0 0 0 1 1	
0	

Problem B

Mission Impossible

Source file name: `mission.c`, `mission.cpp`, `mission.java` or `mission.pas`

You have been hired to explore enemy territory. It is risky business, you know that. So, you'd better be prepared! The enemy has placed a number of security points all over his country, from which radars are detecting any moving vehicle within their range of cover. Any such detected object will be immediately destroyed. Fortunately enough, you have been given by your government a map of the enemy territory, consisting of coordinates and radius of coverage of each radar. You have also a list of local informers (together with their locations) that you should contact in order to obtain valuable information. Your mission is to try to contact one of these informers, preferably the one with highest *insider-coefficient*. The insider-coefficient of each informer is simply the distance from the informer to the border of the country, where such a distance is defined as the minimum over all distances from the location of the informer to each point of the border. In intuitive sense, the informer with highest insider-coefficient is that who is located as inside the country as possible, and will presumably have more valuable information about the country.

Your first thought is then to design a computer program which will check if there is a path from your initial location, always the point (2000,2000), to any of the informers' location, without crossing any region which is covered by radar. Whenever possible, the program should indicate which reachable informer is the one to be contacted, according to the insider-coefficient criteria described above.

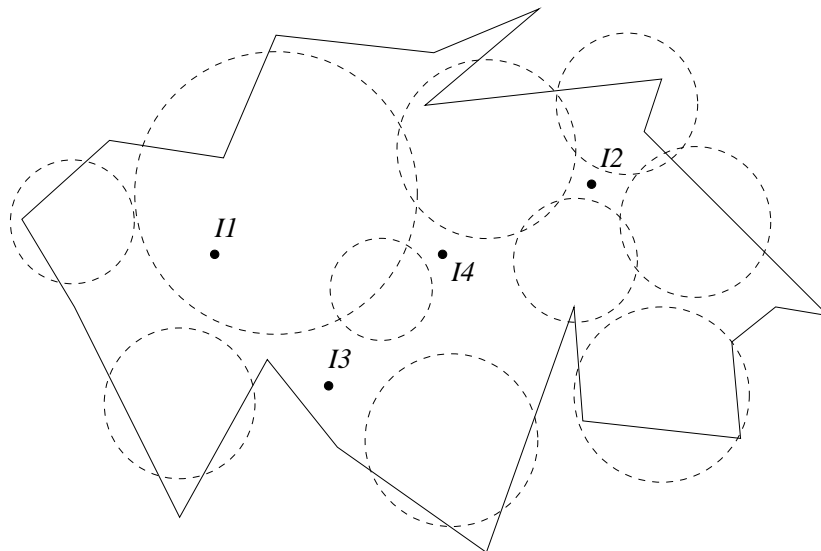


Figure 1: Possible Scenario

The enemy country has the shape of a *simple polygon* (*not necessarily convex*). Recall that a polygon is called simple if it is described by a single, non-intersecting boundary. The borders of the country will be given as a sequence of X,Y-coordinates corresponding to the sequence of vertexes of the polygon. You may assume that all the radar's centres and the informers' coordinates are located within the country's border. Notice, however, that the area covered by the radars might include regions outside the border.

In the sample scenario of Figure 1, informer $I1$ cannot be contacted since he is inside the region covered by radars. The informer $I2$, although outside the radar's region, can't be contacted either since any trip to his location would go through the deadly radar-covered regions. Both informers $I3$ and $I4$ could be contacted, so that informer $I4$ is chosen since his insider-coefficient is greater than that of $I3$.

Input

The input consists of several test cases. The first line of each test case describes the border of the enemy country, in the format

$B \ X1 \ Y1 \ X2 \ Y2 \ \dots \ XB \ YB$

where $3 \leq B \leq 1000$ is the number of border points, and each $X_i \ Y_i$ is the coordinate of the i -th point in the border. The border of the country consists of line segments between points i and $i + 1$, and between points B and 1. The second line gives the number of informers and their respective positions, in the format

$N \ X1 \ Y1 \ X2 \ Y2 \ \dots \ XN \ YN$

where $1 \leq N \leq 1000$ is the number of informers, and $X_i \ Y_i$ is the coordinate of the i -th informer. The third line describes the position and radius of the radars, in the format

$M \ X1 \ Y1 \ R1 \ X2 \ Y2 \ R2 \ \dots \ XM \ YM \ RM$

where $1 \leq M \leq 30$ is the number of radars, $X_i \ Y_i$ is the coordinate of the i -th radar, and R_i is the radius of the i -th radar. All the coordinates are integers $0 \leq X, Y \leq 1000$. The radius of the radars are integers in the range $1 \leq R \leq 1000$. A test case where $B = N = M = 0$ indicates the end of the input. This test case must not be processed.

The input must be read from standard input.

Output

For each test case in the input, your program must produce one line containing either “Mission impossible” or “Contact informer K”, where “K” is the index of the informer (as given in the input) with highest insider-coefficient which can be reached by the spy without going inside any radar coverage area. If there are more than one informer satisfying this condition, choose the one among them with lowest index.

The output must be written to standard output.

Sample input	Output for the sample input
4 0 0 0 200 200 200 200 0	
2 70 70 120 120	
1 100 100 100	
4 0 0 0 200 200 200 200 0	
3 100 102 70 80 20 10	
4 70 70 35 130 70 35 130 130 35 70 130 35	
0	
0	
0	
	Mission impossible
	Contact informer 3

Problem C

Pascal Library

Source file name: pascal.c, pascal.cpp, pascal.java or pascal.pas

Pascal University, one of the oldest in the country, needs to renovate its Library Building, because after all these centuries the building started to show the effects of supporting the weight of the enormous amount of books it houses.

To help in the renovation, the Alumni Association of the University decided to organize a series of fund-raising dinners, for which all alumni were invited. These events proved to be a huge success and several were organized during the past year. (One of the reasons for the success of this initiative seems to be the fact that students that went through the Pascal system of education have fond memories of that time and would love to see a renovated Pascal Library.)

The organizers maintained a spreadsheet indicating which alumni participated in each dinner. Now they want your help to determine whether any alumnus or alumna took part in all of the dinners.

Input

The input contains several test cases. The first line of a test case contains two integers N and D indicating respectively the number of alumni and the number of dinners organized ($1 \leq N \leq 100$ and $1 \leq D \leq 500$). Alumni are identified by integers from 1 to N . Each of the next D lines describes the attendees of a dinner, and contains N integers X_i indicating if the alumnus/alumna i attended that dinner ($X_i = 1$) or not ($X_i = 0$). The end of input is indicated by $N = D = 0$.

The input must be read from standard input.

Output

For each test case in the input your program must produce one line of output, containing either the word ‘yes’, in case there exists at least one alumnus/alumna that attended all dinners, or the word ‘no’ otherwise.

The output must be written to standard output.

Alumna: a former female student of a particular school, college or university.

Alumnus: a former male student of a particular school, college or university.

Alumni: former students of either sex of a particular school, college or university.

Sample input	Output for the sample input
3 3	yes
1 1 1	no
0 1 1	
1 1 1	
7 2	
1 0 1 0 1 0 1	
0 1 0 1 0 1 0	
0 0	

Problem D

ICPC Strikes Again

Source file name: icpc.c, icpc.cpp, icpc.java or icpc.pas

International Concrete Projects Company (ICPC) is a construction company which specializes in building houses for the high-end market. The company is the most profitable company in the world due to a very efficient land division method which has been used in its housing development projects since last year. Recently there was a chaos at ICPC, because employees refused to work arguing that they did not earn enough. Worried about the loss in profit due to the strike, the company board proposed a new method to calculate the salaries which was luckily accepted by everyone.

The salary of a worker reflects the significance of the tasks that he/she has to perform and is influenced by the way tasks depend on each other.

A task X *depends* on a task Y if either (i) X *depends directly* on Y , or (ii) there exists a task T such that X depends directly on T and T depends on Y . Since in ICPC all tasks must be performed, there is no circularity in the task dependence relation. Also, a task may be performed by more than one worker.

A *basic significance* is associated with each task reflecting its importance (for example, developing the efficient land division method is more important than building the houses themselves). The *significance* of a task T is then defined as the basic significance of T plus the significance of every task which depends directly on T . Note that if no other tasks depend directly on task T , the basic significance and the significance of T are the same.

The salary of a worker is the sum of the significances of all the tasks he/she performs which do not depend on any other task performed by him/her. In other words, a value equal to the significance of task X will be added to the salary of a worker W that works in task X if there is no other task Y on which X depends, and W works also in Y .

ICPC wants you to help them to determine the salary of each of its employees.

Input

The input contains several test cases.

The first line of a test case contains two integers T and E indicating respectively the number of tasks and the number of employees ($1 \leq T \leq 1000$ and $1 \leq E \leq 1000$). Tasks are numbered from 1 to T and employees from 1 to E .

Then it will come a sequence of lines describing the tasks 1 to T in ascending order. Each task is described by two lines. The first of these lines contains three integers BS , ND and NE , representing respectively the basic significance of the task, the number of tasks that depend directly on it, and the number of employees who perform it ($1 \leq BS \leq 1000$, $0 \leq ND < T$ and $1 \leq NE \leq E$). The second line contains $ND + NE$ integers corresponding first to the ND directly dependent tasks and then the NE employees who perform the task.

The end of input is indicated by $T = E = 0$.

The input must be read from standard input.

Output

Test cases must be answered in the order that they were presented. For each test case you must print:

- a single line containing five stars ***** indicating the beginning of the case
- for each employee i , one line with two integers i and s , separated by a blank, meaning that i has a salary of s .

The output must be written to standard output.

Sample input	Output for the sample input
3 2	*****
100 2 2	1 200
2 3 1 2	2 200
40 0 1	*****
1	1 70
60 0 1	2 60
2	
7 2	
10 2 1	
2 3 1	
10 2 1	
4 5 2	
10 2 1	
6 7 2	
10 0 1	
1	
10 0 1	
1	
10 0 1	
1	
10 0 1	
1	
0 0	

Problem E

Optical Fiber

Source file name: fiber.c, fiber.cpp, fiber.java or fiber.pas

A far away developing country is trying to improve its communication infrastructure. Currently, each city in the country has its local computer network, but there is no fast communication between the cities. The Autonomous Communications Ministry (ACM) of the country decided to create a fast, optical fiber network connecting every city. In order to do this, they decided to take the following approach. Pairs of cities were chosen to have an optical fiber link installed between them. The choice was such that there will be only one fiber path between any pair of cities, in order to reduce the cost. The pairs of cities were chosen considering many factors, including analysis of estimated demand and distance between the cities.

Each city will have one optical router installed, which will be used to connect all the optical links with one end in that city. In each city, there are many different locations where the optical router can be installed. Your task, as an engineer working on this project, is to develop a computer program to pick up the locations in each city in order to minimize the total length of fiber that will be necessary for this project.

Input

The input consists of multiple test cases. Each test case starts with a line containing the number of cities N ($1 \leq N \leq 1000$) in the country. Following, for each city, there is a sequence of lines. The first line contains the (unique) name of the city (only capital letters, at most 15), and the number of candidate sites C_i ($1 \leq C_i \leq 50$) where the optical router can be installed. Then, there is one line per candidate site, containing two integers X and Y representing the coordinates of the site ($-10000 \leq X, Y \leq 10000$). You should use the euclidean distance between the sites to compute the corresponding fiber length necessary to connect them. After the description of each city with its candidate sites, there are $N - 1$ lines, each containing the names of two cities that will have a fiber link installed between them. The end of the input is indicated by $N = 0$.

The input must be read from standard input.

Output

For each test case, your program should print a line with the minimum total length of optical fiber to connect the requested cities. Your answer should be rounded to one decimal digit.

The output must be written to standard output.

Sample input	Output for the sample input
3 AUSTIN 1 500 500 DALLAS 2 1000 10 990 -10 ELPASO 2 0 0 30 0 ELPASO AUSTIN DALLAS ELPASO 3 HUSTON 3 100 0 100 50 100 100 AUSTIN 2 200 0 180 40 SANANTONIO 2 0 -10 10 -50 HUSTON AUSTIN HUSTON SANANTONIO 0	1646.3 189.9

Problem F

Where Are My Genes

Source file name: genes.c, genes.cpp, genes.java or genes.pas

One way that scientists try to measure how one species evolved into another is to find out how the ancestor's genome changed into the other's. Closely related species have several genes in common and it turns out that a good way to compare them is comparing how the common genes changed place.

One of the most common mutations that change the order of a genomes' genes is the reversal. If we model a genome as a sequence of N genes with each gene being an integer from 1 to N , then a reversal is a mutation that changes the genome by reverting the order of a block of consecutive genes. A reversal can be described by two indexes (i, j) , $1 \leq i \leq j \leq N$, indicating that it reverts the order of the genes within indexes from i to j . So, when it is applied to a genome $[g_1, \dots, g_{i-1}, g_i, g_{i+1}, \dots, g_{j-1}, g_j, g_{j+1}, \dots, g_N]$, we obtain the genome $[g_1, \dots, g_{i-1}, g_j, g_{j-1}, \dots, g_{i+1}, g_i, g_{j+1}, \dots, g_N]$. As an example, the reversal $(3, 6)$ applied to the genome $[1, 2, 3, 4, 5, 6, 7]$ gives $[1, 2, 6, 5, 4, 3, 7]$. If after that the reversal $(1, 3)$ is applied, we obtain the genome $[6, 2, 1, 5, 4, 3, 7]$.

A scientist studying the evolution of a species wants to try a series of reversals on its genome. Then he wants to query the final position of several genes. Would you take the challenge and help him?

Input

The input contains several test cases. The first line of a test case contains one integer N indicating the number of genes in the genome ($1 \leq N \leq 50000$). You may assume that the initial order of the genes is the sequence of integers from 1 to N in increasing order. The second line of a test case contains one integer R ($0 \leq R \leq 1000$) indicating the number of reversals to be applied to the genome. Then R lines follow, each containing two integers i, j ($1 \leq i \leq j \leq N$), separated by a single space, indicating the two indexes that define the corresponding reversal. After the description of the reversals there is a line containing one integer Q ($0 \leq Q \leq 100$), indicating the number of queries for genes, followed by Q lines, where each line contains an integer representing a gene whose final position you must determine.

The end of input is indicated by $N = 0$.

The input must be read from standard input.

Output

For each test case in the input your program must produce $Q + 1$ lines of output. The first line must contain the string "Genome " followed by the number of the test case. The following Q lines must contain one integer each representing the answers of the queries.

The output must be written to standard output.

Sample input	Output for the sample input
9 1 3 6 4 1 3 5 1 5 2 1 2 1 5 2 5 2 0	Genome 1 1 6 4 1 Genome 2 1 5

Problem G

Computer DJ

Source file name: dj.c, dj.cpp, dj.java or dj.pas

A very famous DJ has been recently invited to play in the closing party of a Computer Science conference. Trying to impress the participants of the event, he decided to use a program in order to choose the songs he would play at this party. However, the result was a disaster, since the way the program chose the songs was quite weird and repetitive.

First of all, the DJ has selected N songs among the set of songs he had available. The program used by the DJ then labels each of the songs using one different character from 'A' to 'Z'. The i^{th} song is labeled using the i^{th} character of the sequence 'A'-'Z'. The program chooses the songs to be played in the party in the order that their labels appear in the following infinite sequence of characters: first come all the words with one character in lexicographical order; then all the words with two characters in lexicographical order; then all the words with three characters in lexicographical order; and so on. For $N = 3$, this sequence would be ABCAAABACBABBCCACBCCAAAAABAACABAABBABC...

At the end of the party, some people asked the DJ if he remembered which the first song played was. Others would like to know which the 25th was, and so on. The DJ remembers nothing but the strange pattern of repetition of the songs, so he urges you to help him and write a program which answers such queries.

Input

The input contains several test cases. Each test case consists of three lines. The first line of a test case contains two integers N and Q indicating respectively the number of songs chosen by the DJ and the number of queries made by the participants ($1 \leq N \leq 26$ and $1 \leq Q \leq 1000$). In the second line, there will be the N titles of the songs (the title of a song is a chain of alphanumerical characters of at least one and at most 100 characters) separated by single spaces. The last line of a test case contains a sequence of queries. Each query is a number k ($1 \leq k \leq 100000000$) corresponding to the k^{th} song played in the party. The end of the input is indicated by $N = Q = 0$.

The input must be read from standard input.

Output

For each query number k in a test case, you shall print a single line containing the name of the k^{th} song played in the party. A blank line must follow each test case.

The output must be written to standard output.

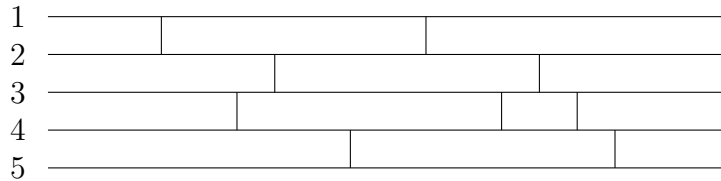
Sample input	Output for the sample input
10 3 S0 S1 S2 S3 S4 S5 S6 S7 S8 S9 3 6 10 3 5 Pathethique TurkishMarch Winter 1 2 3 4 16 0 0	S2 S5 S9 Pathethique TurkishMarch Winter Pathethique Winter

Problem H

P-Networks

Source file name: pnets.c, pnets.cpp, pnets.java or pnets.pas

Pretty Networks Inc. is a company that builds some curious artifacts whose purpose is to transform a set of input values in a given way. The transformation is determined by what they call a p-network. The following picture shows an example of a p-network.



In the general case, a *p-network of order N and size M* , has N horizontal wires numbered $1, 2, \dots, N$, and M vertical strokes. Each stroke connects two consecutive wires. There are no two different strokes touching the same point of any wire, and there is no stroke touching the leftmost or rightmost point of any wire. The above example is a p-network of order 5 and size 9.

The transformation determined by a p-network can be explained using a set of rules that govern the way a p-network should be traversed:

1. start at the leftmost point of one wire, and go to the right;
2. each time a stroke appears move to the connected wire, and keep going from left to right;
3. stop when the rightmost point of one wire is reached.

If starting at wire i the traversing ends at wire j , we say that the p-network transforms i into j , and we denote this with $i \rightarrow j$. In the above example the p-network determines the set of transformations

$$\{1 \rightarrow 3, 2 \rightarrow 5, 3 \rightarrow 4, 4 \rightarrow 1, 5 \rightarrow 2\} .$$

Pretty Networks Inc. hired you to solve the following p-network design problem: given a number N and a set of transformations $\{1 \rightarrow i_1, 2 \rightarrow i_2, \dots, N \rightarrow i_N\}$, decide if a p-network of order N can be built to accomplish them and, in this case, give one that does it.

When there exists a solution with a certain size, in many cases there is another solution with a greater size. Scientists at Pretty Networks Inc. have stated that if there exists a solution for a p-network design problem, then there is a solution with size less than $4N^2$. Therefore, they are interested only in solutions having a size below this bound.

Input

The input has a certain number of p-network design problems. Each problem is described in just one line that contains the values N, i_1, i_2, \dots, i_N , separated by a single blank space. The value N is the order of the desired p-network, i.e., its number of wires ($1 \leq N \leq 20$). The values i_1, i_2, \dots, i_N represent that the p-network should determine the set of transformations $\{1 \rightarrow i_1, 2 \rightarrow i_2, \dots, N \rightarrow i_N\}$ ($1 \leq i_j \leq N$, for each $1 \leq j \leq N$). The input ends with a line in which $N = 0$; this line must not be processed as a p-network design problem.

The input must be read from standard input.

Output

For each p-network design problem in the input, the output must contain a single line. If the problem has no solution the line must be **No solution**. Otherwise the line must contain a description of any p-network (with N wires and less than $4N^2$ strokes) that accomplishes the requested set of transformation. This description is given by a set of values M, s_1, s_2, \dots, s_M , where consecutive values are separated by a single blank space. The value M is the size of the p-network, i.e., its number of strokes. The values s_1, s_2, \dots, s_M describe the strokes of the p-network; it should be understood that the i -th stroke from left to right, connects the wires s_i and $1 + s_i$ ($1 \leq i \leq M$). Notice that $0 \leq M < 4N^2$, while $1 \leq s_i < N$ for each $1 \leq i \leq M$.

The output must be written to standard output.

Sample input	Example of output for the sample input
<pre>5 3 5 4 1 2 3 1 1 3 2 1 2 2 1 2 0</pre>	<pre>9 1 3 2 4 1 3 2 3 4 No solution 0 2 1 1</pre>

Problem I

Light Up

Source file name: `light.c`, `light.cpp`, `light.java` or `light.pas`

Light Up is a puzzle set in a rectangular board divided in smaller squares. Some squares in the board are “empty” (white squares the figure below), some squares are “barriers” (dark squares in the figure below). A barrier square may have an integer number i associated to it ($0 \leq i \leq 4$).

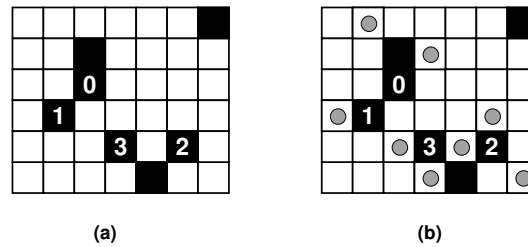


Figure 2: (a) Puzzle with 6 rows, 7 columns and 7 barriers; (b) a solution to the puzzle.

In this puzzle the goal is to “light up” all the empty squares by placing lamps in some of them (lamps are depicted as circles in the figure). Each lamp illuminates the square it is on, plus all squares in line with it, horizontally or vertically, up to a barrier square or the board end.

A winning configuration satisfies the following conditions:

- all empty squares must be lit;
- no lamp may be lit by another lamp;
- all numbered barrier squares must have exactly that number of lamps adjacent to them (in the four squares above, below, and to the side);
- non-numbered barrier squares may have any number of lamps adjacent to them.

You must write a program to determine the smallest number of lamps that are needed to reach a winning configuration.

Input

The input contains several test cases. The first line of a test case contains two integers N , M indicating respectively the number of rows and the number of columns of the board ($1 \leq N \leq 7$, $1 \leq M \leq 7$). The second line contains one integer B indicating the number of barrier squares ($0 \leq B \leq N \times M$). Each of the next B lines describe a barrier, containing three integers R , C and K , representing respectively the row number ($1 \leq R \leq N$), the column number ($1 \leq C \leq M$) and the barrier number ($-1 \leq K \leq 4$); $K = -1$ means the barrier is unnumbered. The end of input is indicated by $N = M = 0$.

The input must be read from standard input.

Output

For each test case in the input your program must produce one line of output, containing either an integer indicating the smallest number of lamps needed to reach a winning configuration, in case such a configuration exists, or the words 'No solution'.

The output must be written to standard output.

Sample input	Output for the sample input
2 2	2
0	No solution
2 2	8
1	
2 2 1	
6 7	
7	
2 3 -1	
3 3 0	
4 2 1	
5 4 3	
5 6 2	
1 7 -1	
6 5 -1	
0 0	