Different time limits will be used in the judging of submissions for different problems. Each time limit has been chosen to be way more than that of the slowest solution developed by the judging team. The value of the "Time Limit" is NOT meant to reflect the difficulty of solving the problem.

	Problem Name	**Time Limit (in seconds)**
A	SecurePINs	2
B	The Large Family	2
C	Start-up Network	15
D	Clock Splitter	2
E	Neocortex order	20
F	Some Pretty Peculiar Cells	3
G	Mealy's Memoirs	2
H	Ninja Pizza	3
I	Count the Squares	15
J	Yet Another Newspaper Puzzle	5
K	Encyclopaedia	10
L	Solar Powered Ants	10

Your program should read from standard input and print into standard output.

Good luck and have fun.

DOMJudge messages and their meaning

For each submission the contest system DOMJudge with respond with one of the following results:

CORRECT
The submission passed the judge's test: you solved this problem!

COMPILER-ERROR
There was an error when compiling your program.
Check your code and make sure the right language has been selected.

TIMELIMIT
Your program took longer than the maximum allowed time for this problem. Therefore it has been aborted. This might indicate that your program hangs in a loop or that your solution is not efficient enough.

RUN-ERROR
There was an error during the execution of your program. This can have a lot of different causes like division by zero, incorrectly addressing memory (e.g. by indexing arrays out of bounds), trying to use more memory than the limit, etc.
Also check that your program exits with exit code 0!

NO-OUTPUT
Your program did not generate any output.
Make sure that you write to standard out.

WRONG-ANSWER
The output of your program was incorrect. This can happen simply because your solution is not correct, but remember that your output must comply exactly with the specifications of the jury.

TOO-LATE
Bummer, you submitted after the contest ended! Your submission is stored but will not be processed any more.

Problem A
Secure PINs

Time Limit: 2 seconds

Most people are aware of the importance of selecting a good password for their computer and e-mail accounts. However, they pay little attention when they choose their credit card and bank card PINs, even though they can probably unlock a lot of wealth.

Your task is to write a program to assess the security level of a collection of PINs. A PIN that contains the same digit three times or a sequence of three consecutive digits (such as 345 and 654) is to be assessed as weak. Otherwise, it is considered acceptable. Note that sequences that wrap around, like “391312” and “098165”, are not considered consecutive.

Input

The input starts with an integer N, on a line by itself, that represents the number of test cases. $1 \leq N \leq 1000$. The description for each test case consists of a six-digit non-negative number on a line by itself.

Output

The output consists of a single line, for each test case, which contains your program's assessment of the PIN as “WEAK” or “ACCEPTABLE”.

Sample Input	Output for the Sample Input
9	ACCEPTABLE
024578	ACCEPTABLE
248905	ACCEPTABLE
391312	ACCEPTABLE
098245	WEAK
145698	WEAK
212324	WEAK
986541	ACCEPTABLE
968541	ACCEPTABLE
540872	

Problem B
The Large Family

Time Limit: 2 seconds

The “Occupy the Kitchen” movement by the **five** children of the Large family, has taken Mr. and Mrs. Large by surprise. The children raised banners demanding the equal distribution of mangos after a large number of mangos went missing from the fridge overnight. They presented their demands as the family food constitution.

The Large Family Food Constitution

- A mango may only be divided into two halves.
- Children shall receive equal amounts of mangos.
- Mum and Dad shall receive equal amounts of mangos.
- Allocation may differ by no more than half a mango amongst all family members.
- Each family member must receive the maximum amount possible without violating the above constitutional items.
- Extra amounts of mangos that result in the violation of the above constitutional items shall go to the goat. Mangos going to the goat must be kept to a minimum.

Mr. and Mrs Large have agreed to the constitution on the condition that software is developed to compute the correct allocation of mangos. Your task is to write a program that allocates mangos to each family member and the goat.

Input

The input consists of many test cases. The description for each test case consists of an integer, $0 < G \leq 300$, on a line by itself that represents the total number of mangos to be divided.

A line with a single zero indicates the end of data and should not be processed.

Output

The output consists of a single line, for each test case, which contains four values separated by single spaces. The first is the number of mangos for the goat, the second is the number of mangos for dad, the third is the number of mangos for mum and the fourth is the number of mangos for each child. Each value is written as a decimal with a single digit after the decimal point.

Sample Input	Output for the Sample Input
41 0	0.0 5.5 5.5 6.0

Problem C
Start-up Network

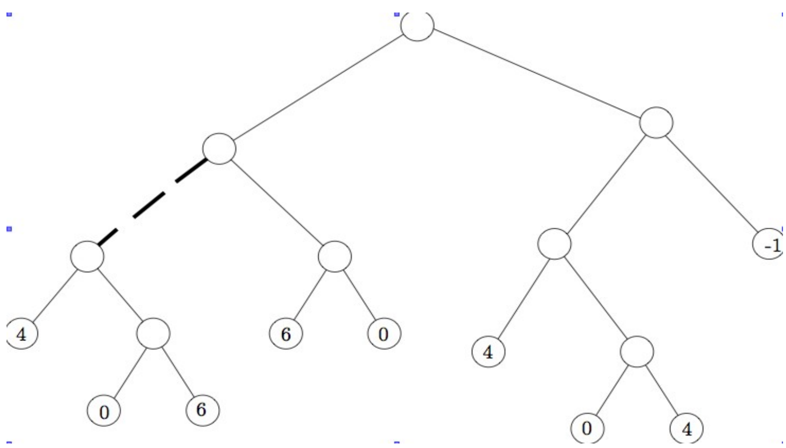
Time Limit: 15 seconds

A start-up communication company plans to build a proprietary network using cheaply available 3-way routers[1] from a local supplier. The network will be configured in a tree-like architecture with all the customers as its leaf nodes.

Customers may be engaged in one-to-one calls or in conference calls but a customer cannot be simultaneously engaged in more than one call. An active one-to-one call uses the links on the unique path between the two leaf nodes corresponding to the two customers. An active conference call uses the links of the connected sub-tree that spans leaf nodes corresponding to the participating customers.

The company is worried about the negative effect that too many active calls using a single link may have on the overall performance of its network. The company wants you to calculate the maximum number of active calls using any link (that is, any edge of the tree) in the network for a given set of calls.

The dashed link in the example network, shown below,



is used by the three active calls, which is the maximum value.

1 A *router* is a device that forwards data packets between two, or more, networks. A router is said to be 3-way if it only has three communication ports.

Input

The input consists of a number of test cases, where the data for each case is a full binary rooted tree[2] of T leaf nodes on a line by itself. A tree is described by a set of matched parentheses and a set of integers in the range of "-1" to "T-1", $3 \le T \le 5000$. Parentheses and integers are separated by single spaces.

A pair of matching parentheses represents the root of one full binary rooted tree and an integer represents a leaf node in the tree. Leaf nodes with the same integer value correspond to the subset of customers engaged in an active call. A leaf node with a value of "-1" indicates a customer who is not engaged in a call.

Two matched parentheses, separated by a single space, on a line by themselves indicate the end of input data and should not be processed.

Output

For each test case, the output consists of a single line. The line begins with the word "Tree" followed by a space, an integer N, the character sequence ":␣", and then an integer indicating the maximum communication load on any tree edge. N is the number of the test case starting with the value "1", and the symbol "␣" indicates a single space.

Sample Input	Output for the Sample Input
((0 1) ((1 1) 0)) ((0 (0 3)) (((1 0) (1 0)) 3)) (((4 (0 6)) (6 0)) ((4 (0 4)) -1)) ( )	Tree 1: 2 Tree 2: 2 Tree 3: 3

2 A *full binary rooted tree* consists of a single leaf node , or a root node connected to two full binary rooted trees. In such a tree, each node has either zero or two child nodes.

Problem D
Clock Splitter

Time Limit: 2 seconds

An analogue clock has the first twelve natural numbers placed in an equally spaced fashion in a clockwise direction on its face. It is possible to draw one line across this regular clock such that the sum of the numbers on both sides of the line are equal, as shown in figure A.

It is not possible to draw such a line for a clock with the first five natural numbers on its face. However it is possible to draw a line such that the sum of the numbers on both sides of the line differ by one, as shown in figure B.

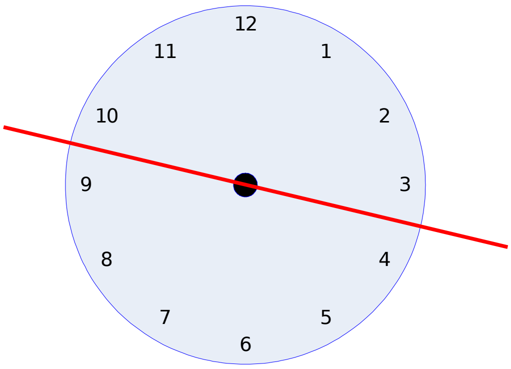


Figure A

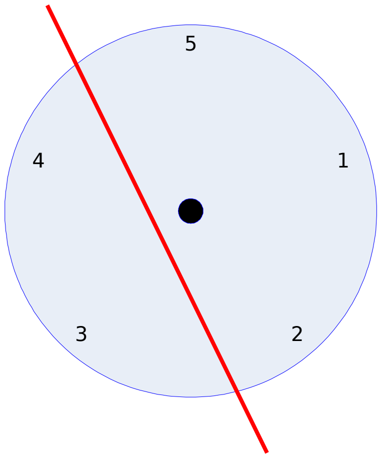


Figure B

Your task is to write a program to find where the line can be drawn for a clock with the first *N* natural numbers such that the sum of the numbers on both sides of the line are as close as possible to each other.

For some values of *N*, there will be more than one possible solution. Your program must report only the line with the smallest starting number.

Input

The input contains a number of test cases with one input value N per case, $2 \leq N \leq 100000$, which is the largest value on the clock face.

A value of zero (0) on a line by itself indicates the end of input and should not be processed.

Output

The output consists of a single line, for each test case, which contains two integers separated by a single space. The integers represent the smallest number followed by the largest number, in the clockwise direction, on the same side of the splitting line.

Sample Input	Output for the Sample Input
12 5 13 0	4 9 3 4 1 9

Problem E
Neocortex Order

Time Limit: 20 seconds

The surface of the neocortex is partitioned into regions, not unlike the map of the Holy Roman Empire circa 1789. The neocortex in each region, which is approximately 3 mm thick, has six layers. The regions are connected by bundles of fibres that form the white matter of the brain. Each fibre starts in one region and sends signals in one direction to another region where it terminates. The starting and terminating layers may be different.

The connectivity from a region P to a region Q, which we shall describe by P→Q, is classified into one of the following three classes:
* Ascending 'A' if fibres from P terminate in layer four of Q.
* Descending 'D' if fibres from P terminate in layers of Q other than layer four.
* Lateral 'L' if fibres from P have terminations in all layers of Q.

For the visual and auditory cortices, it should be possible to arrange the regions into a hierarchy of discrete levels, such that for regions P and Q:

- P and Q are placed on the same level if either P→Q or Q→P are classified as 'L'.
- P is placed on a strictly lower level than Q if P→Q is classified as 'A' or if Q→P is classified as 'D'.
- P is placed on a strictly higher level than Q if P→Q is classified as 'D' or if Q→P is classified as 'A'.

Due to inconclusive experimental data for a given neocortex, the resulting classification of a number of connections may be in disagreement with any possible hierarchy. An 'L' connection is in disagreement with a hierarchy if the regions are not on the same level, an 'A' connection is in disagreement if the first region is not on a strictly lower level than the second region, and a 'D' connection is in disagreement if the first region is not on a strictly higher level than the second region.

Researchers have observed that no more than **five (5)** disagreements may occur. However they would like as few disagreements as possible to be removed to allow for the regions to be arranged into a hierarchy. Your task is to write a program to determine the minimum such number.

Input

The input consists of several test cases. Each test case starts with an integer N, $1 \leq N \leq 25$, on a line by itself.

Each of the following N lines consists of three parts: a string STR1, followed by a letter T and then another string STR2. The three parts are separated by single spaces, and the two strings consist of letters and digits. The STR1 and STR2 correspond to two different regions, say r1 and r2. The letter T has one of the values 'A', 'D', 'L' to indicate the connection type from r1 to r2. Each line is unique within a test case.

A zero on a line by itself declares the end of input data and should not be processed.

Output

The output consists of a single line, for each test case, which contains the smallest number of connections in disagreement, over all possible arrangements of the regions into a hierarchy.

Sample Input	Output for the Sample Input
7 V2 A MT MT D V2 V2 A V4 V4 D V2 V4 A MT V4 L MT MT L V4 7 V2 A MT MT D V2 V2 A V4 V4 D V2 V4 A MT V4 D MT MT L V4 2 MT A MSTd 36 D 7a 0	1 2 0

Problem F
Some Pretty Peculiar Cells

Time Limit: 3 seconds

Samantha is a graduate student who has been studying a new variety of cells, which she named SPP (Some Pretty Peculiar) Cells. Samantha's first observations of SPP cells led her to believe that they are benign. That is, they neither increase nor decrease their number over time. Further investigations, by a co-researcher who wanted to duplicate her results, showed SPP cells to be a very aggressive type of cells that double their number every minute. Now Samantha has to explain her conclusions about SPP cells being benign to her supervisor who is very unhappy about the time he has wasted on a grant application based on her results.

Samantha has a theory to explain her data, which goes as follows:

> Each SPP cell occupies exactly one (1) square micrometre of space in the rectangular Petri dish. As the population of SPP cells grows exponentially fast, they very quickly occupy all the space in the Petri dish (i.e., there are as many cells as the area of the dish). When this happens, a large number of the cells die from overcrowding.
>
> The way the cells divide and die is very unique according to the theory. When the space in the dish does not allow for all the cells to divide, a number of cells go into hibernation for one (1) minute. Each of the remaining cells duplicates, which makes the total number of cells equal to the Petri dish's area. The newly born cells then die immediately from overcrowding. In the next minute, all the hibernating cells wake up and continue their exponential growth.
>
> For example, consider a rectangular Petri dish with an area of 100,000 square micrometres that contains 30,000 SPP cells at the starting time T = 0. At 1 minute, every cell splits and there will be 60,000 SPP cells. In the second minute 40,000 SPP cells will duplicate and 20,000 will hibernate, which brings the total number of cells to 100,000. The 80,000 cells that were just born will die, from overcrowding, leaving just 20,000 SPP cells at the beginning of the second minute. At the third minute the disc will have 40,000 SPP cells, and so on.

Your task is to write a program to verify Samantha's theory. Your program should report the smallest interval of time T, where T > 0, at which time the Petri dish contains the same number of SPP cells as it has at time T = 0. If there is no such interval, the program should report IMPOSSIBLE!

Input

The input starts with an integer N, on a line by itself, that represents the number of cases. $1 \leq N \leq 1000$. The description for each case consists of three integers, X, Y and A, separated by single blank spaces, on a line by themselves. X and Y represent the length of the rectangular Petri dish sides in micrometres, and A is the starting number of SPP cells in the Petri dish. $1 \leq X, Y, A \leq 40000$.

Output

The output consists of a single line, for each test case, which contains the shortest interval of non-zero time until the number of SPP cells is the same as it was at time 0. If there is no such interval print the string "IMPOSSIBLE!"

Sample Input	Output for the Sample Input
3 3 4 8 3 7 2 10000 10000 1	2 6 IMPOSSIBLE!

Problem G
Mealy's Memoirs

Time limit allowed: 2 seconds

After a distinguished career as a computer scientist, Reginald Mealy wrote his memoirs and used a speech recognition program to transcribe his life story as he spoke. The result was many large streams of text.

However, all did not go according to plan because the memoirs contained the email addresses of many of his associates. The speech recognition programs transcribed the email addresses literally such that me@me.com was transcribed as meatmedotcom. All special characters were transcribed literally: "@" was transcribed as "at" and "." was transcribed as "dot". As well, spaces were omitted unless there was an unusually long pause.

Reginald Mealy wants to extract the email addresses from the transcribed text, and he has set it as a task for his student. As an incentive, he has offered a bounty of 128 cents for each extracted email address.

The student, being greedy, would like to maximise the number of extracted email addresses, and would like you to write a program to calculate the maximum bounty that can be collected for a stream of text. You may assume that all email addresses have one of the following forms:

[word]@[word].[word]
[word]@[word].[word].[word]
[word]@[word].[word].[co]
[word]@[word].[word].[word].[co]

where

[word] is a word of between 3 and 16 lower case Roman letters, and
[co] is a two letter country code,
where all two-letter combinations are valid country codes.

So the following two strings aaa@aaa.aaa and aaa@aaa.aaa.aaa.aa are valid email addresses, but the following three strings a@a.a, @a or a@a. are not valid. Note that a bounty is collected only once if multiple email addresses overlap.

Input

The input starts with an integer N, on a line by itself, which indicates the number of test cases. $1 \leq N \leq 100$. The description of each test case consists of a string made entirely of lower case Roman letters from a to z, inclusive, and the space character.

Output

For each test case the output consists of a single line that contains the maximum bounty that can be received. The amount is to be given in the format of $d.cc, where d is the number of dollars and cc is the number of cents (with a leading 0 if required).

Sample Input
6 aaaaaaaameatmeedotcombbbbbbbb bbbbmattatdiedotdotnetpatdotdotherityatiinetdotnetaaaa myemailathomedot com myemailatabcdefghijklmnopqrstuvwxyzdotcom helmattatdidotdotnet helmatdotdotherityatiinetdotnet

Output for the Sample Input
$1.28 $3.84 $0.00 $0.00 $1.28 $2.56

Problem H
Ninja Pizza

Time Limit: 3 seconds

Ninja Pizza is getting popular because its delivery boys wear ninja costumes pretending to be real ninjas who are renowned for their excellent sword skills. The company advertisements say that a ninja can easily slice two of their perfectly round pizzas, when placed with their interiors non-overlapping on a flat table, into two equal-area halves with one swing of their sword.

The delivery boys are not real ninjas and the shape of each pizza is not really round, but is in the form of a convex polygon with N vertices. $3 \leq N$.

Your task is to write a program that calculates the necessary sword stroke, which a real ninja would instinctively do, for a given two pizzas. A sword stroke is described by a straight line in the form of $y = m * x \pm b$ in the x-y plane.

Input

The input starts with an integer C, on a line by itself, that represents the number of test cases. $1 \leq C$. Each test case contains the specifications of two convex polygons. The first line of a polygon specification consists of an integer N that represents the number of its vertices. Each of the following N lines contains two real numbers, separated by a single space, that are the x- and y-coordinates of one vertex, respectively. The vertices are given in the anti-clockwise order.

The total number of vertices over all test cases is not more than 5000 and values of all x- and y- coordinates are in the range from -1 to 1, inclusive.

There will always be a solution for each case with $-1000 \leq m \leq 1000$.

Output

For each test case the output consists of a single line that contains the equation of a planar straight line that bisects both polygons. The equation is to be printed in the familiar form of "y␣=␣m␣x␣±␣b", where the symbol "␣" indicates a single space and "±" indicates the sign of b.

The values of m and b must be printed as shown in the sample data below, after rounding off to the hundredths place.

Sample Input	Output for the Sample Input
1 4 0.0 0.2 0.1 0.2 0.1 0.3 0.0 0.3 4 0.1 0.3 0.2 0.3 0.2 0.4 0.1 0.4	y = 1.00 x + 0.20

Printing of zero

The value of zero should always be printed as 0.00

An output of "-0.00" will result in a WRONG-ANSWER and 20 minutes penalty.

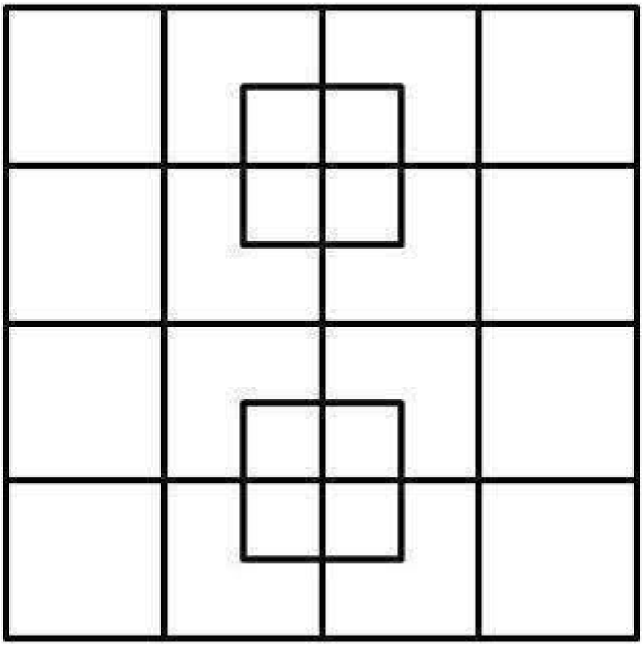
Rounding off is used to approximate the value of a number to a specified decimal place. After rounding, the digit in the place we are rounding will either stay the same, referred to as rounding down, or increase by 1, referred to as rounding up. Examples are:

if we round 6.734 to the hundredths place, it is rounded down to 6.73.
if we round 6.735 to the hundredths place, it is rounded up to 6.74.
if we round -6.734 to the hundredths place, it is rounded up to -6.73.
if we round -6.735 to the hundredths place, it is rounded down to -6.74.

Problem I
Count the Squares

Time Limit: 15 seconds

The puzzle "*How many squares can you count in this image*?"



has recently been popular amongst members of some social networking sites. A posting of such a puzzle is usually followed by an endless number of likes, replies with possible answers, and arguments.

Posting a code that calculates the correct number of squares for such a puzzle would be a great spoiler. Your task is to write a program to solve this type of puzzle.

By the way, the answer for the above puzzle, which is an 8x8 grid, is **40**.

Input

The input starts with an integer N, on a line by itself, that represents the number of puzzles. $1 \le N \le 100$.

The description for each puzzle starts with an integer L, on a line by itself , that represents the total number of line segments that form the puzzle. The set of line segments are rectilinear. That is, its elements are parallel to the x- or y-axis. $0 \le L \le 2000$.

Each of the following L lines contain 4 integers, separated by single blank spaces, that describe a single line segment. The first two integers are the x- and y-coordinates of one end, and the second pair of numbers are the x- and y-coordinates of the other end. Values of the x- and y-coordinates are in the range of 1 to 10000, inclusive.

Output

The output consists of a single line, for each test case, which contains an integer that represents the total number of squares.

Sample Input	Output for the Sample Input
2 9 1 1 1 2 1 2 1 3 1 2 2 2 2 1 2 2 2 2 2 3 2 1 3 1 2 2 3 2 3 1 3 2 3 2 3 3 8 1 1 4 1 1 1 1 4 1 4 4 4 4 1 4 4 3 3 6 3 3 3 3 6 3 6 6 6 6 3 6 6	1 3

Problem J
yet Another Newspaper Puzzle

Time limit allowed: 5 seconds

Some newspapers attempt to increase their sales by offering puzzles that challenge the readers and occupy the time of their daily commute.

Alfakodo is one such puzzle that asks for unique numeric values to be assigned to the twenty-six letters of the English alphabet, subject to given constraints. Each constraint is given as a clue in the form of a simple arithmetic expression. For example, A = N-X.

Since the puzzle is not meant to frustrate the reader, the puzzle is constructed in such a way that throughout the solution process there is always at least one letter whose value can be ascertained.

ALFAKODO

Letters A to Z have a number value
Some are shown in the right hand cells
Create remaining values using clues in centre cells

A	N-X		N	A×L	
B	X+Q	22	O	T+V	17
C	T+L		P	S+A	9
D	O-Q	3	Q	T+D	
E	P×A	18	R	D+E	
F	M+Y		S	R÷D	7
G	N+C		T	H+S	
H	R-O		U	T+F	
I	A+E		V	E÷D	
J	C+D		W	L+N	
K	J+V		X	S+M	
L	K÷L	5	Y	S+L	
M	X-S		Z	S+C	

Input

The input starts with an integer N, on a line by itself, which indicates the number of test cases. $1 \leq N \leq 100$. The description of each test case consists of twenty six (26) lines, one line for each letter of the English alphabet. Each line consists of two or three parts, separated by single spaces:

1. The first part is a letter of the alphabet from A to Z, inclusive.
2. The second part is an arithmetic expression that consists of two letters separated by one of the four operations (+, - , / and *).
3. The third part, if present, is an integer value in the range of one (1) to fifty (70), inclusive.

Output

For each test case the output consists of a single line that contains the values of the alphabet letters, from A to Z, separated by single spaces.

Sample Input
1 A N-K B Z-I C A+N 21 D Y-W E R+P 8 F I+U G Y+I 39 H Q+R I I/R 2 J Q-W 22 K I+P L L*R M H-F N E+P O C-V 5 P P/R Q L-I R H-Q S F+W T Y-B U G-J V V*R W X-C X N+V Y G-I Z C+P

Output for the Sample Input
6 26 21 27 8 19 39 33 2 22 9 34 14 15 5 7 32 1 29 11 17 16 10 31 37 28

Problem K
Encyclopaedia of Equality

Time Limit: 10 seconds

Edith has planned for her “*Encyclopaedia of Equality*” to have the property that all volumes of the encyclopaedia have exactly the same length. As a result, editing one article occasionally raises the need for shuffling of articles between the volumes to restore the equality of lengths. Edith has found that the shuffling of articles makes it impossible to maintain the article names in alphabetically sequential order through the encyclopaedia. For example, a volume may contain the articles from *Cockroach* to *Demagogue* and from *Ethiopia* to *Finger* and an article on *Oysters*, while another volume may contain an article on *Democracy*.

Edith wants to name each volume with the minimum number of alphabetically sorted sequential article ranges, separated by “,␣”. A range is either a single article name, or the first and last article name in the range separated by “␣-␣”. The symbol “␣” indicates a single space. Examples of volume names are:

- *Democracy*
- *Cockroach␣-␣Demagogue,␣Ethiopia␣-␣Finger,␣Oyster*

Edith tried to manually update the volume names when she shuffled some articles, but this proved to be too laborious. Your task is to write a program that updates volume names after a reshuffle of articles.

Input

The first line of input contains two integers M and N, separated by a single space, on a line by themselves. M is the number of articles in the encyclopaedia and N is the number of test cases, where $0 < M < 1000000$ and $0 < N < 10000$.

The second line contains M words separated by single spaces and sorted in increasing alphabetical order. Each word is the name of an article in Edith's encyclopaedia and consists of a sequence of alphabetic characters.

The rest of the input contains a series of N test cases. The first line in each test case begins with the word START followed by an initial volume name. Subsequent lines consist of one of the following options:

- A line starting with ADD followed by the article ranges, separated by “,␣”, to be added to the volume. Articles to be added may already exist in the volume name.
- A line starting with REMOVE followed by the article ranges, separated by “,␣”, to be removed from the volume. Articles are not necessarily included in the volume before they are removed.
- A line that contains the single word END that indicates the end of the test case.

In each ADD or REMOVE line, there are at most 50,000 ranges. The ranges are not necessarily sorted alphabetically but the second article name in a range will always occur after the first article name in alphabetical order. Also the ranges are not necessarily specified minimally but they do not overlap.

Output

The output consists of a single line, for each test case, which contains the properly formatted minimum volume name after all the ADD and REMOVE updates have been performed.

Sample Input
10 2
Cockroach Costume Demagogue Democracy Demonstration Eagle Ethiopia Finger Helicopter Oyster
START Cockroach - Demagogue, Ethiopia - Finger, Oyster
ADD Democracy - Demonstration
REMOVE Oyster
END
START Costume - Eagle
REMOVE Eagle, Costume
END

Output for the Sample Input
Cockroach - Demonstration, Ethiopia - Finger
Demagogue - Demonstration

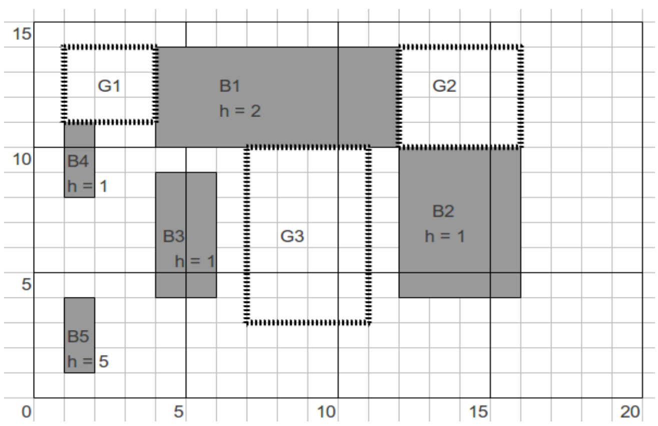
Problem L
Solar Powered Ants

Time Limit: 10 seconds

In Blockland all houses have a rectangular base, vertical walls and a flat roof. Houses, and of course gardens, are all axis-aligned. In addition, local council regulations require that the height of a house be a fixed multiple of the height of the council one-storey building. Blockland is located on the equator of a planet on which the sun always goes from east to west, passing directly overhead at midday. Since there is no seasonal variation, Blocklanders measure time during the day by the length of shadows cast by a one-storey house. They call it shadow time, abbreviated as *ST*. Negative numbers denote shadows to the west (during the morning) and positive numbers denote shadows to the east (during the afternoon). Midday is 0 ST (i.e. zero shadow time).

Once a year, Blockland is invaded from all sides by an army of ants that devour any garden produce they can reach. The ants derive their energy directly from the sun and stop dead if any part of their solar absorbing body is not in direct sunlight. When the arrival hour of the ants is known, Blocklanders need to identify the status of their garden areas as safe or vulnerable to attack. Gardens whose entire areas are not reachable by the ants, due to presence of buildings and/or shadows, are safe and in no need of further protection.

At 2 ST, G1 is sunlit and exposed to attack from the top and the left. G2 is entirely in shadow from building B1 (because that building has a height of 2 and hence shadows of length 4) and therefore safe. G3 is partially sunlit but is protected by buildings B1, B2 and B3 and the shadows of buildings B4 and B5.



Your task is to write a program for Blocklanders that identifies the status of the garden regions.

Input

The input consists of a number of test cases. The description for each test case starts with three integers, *b, g,* and *t*, separated by single spaces on a line by themselves. *b, g,* and *t* are the number of buildings, the number of gardens and the shadow time of the ants invasion, respectively.
$0 \le b, g \le 3000$, and $-1,000,000 \le t \le 1,000,000$.

Each of the following *b* lines contain 5 integers, *x*, *y*, *w*, *b*, *h*, separated by single spaces. *x* and y represent the coordinates of the bottom left corner of a building, *w* and *b* represent the *x*-span and *y*-span of the building, and *h* represents the height of the building in storeys. Buildings do not overlap other buildings. $0 \le x, y \le 1,000,000$, and $1 \le w, b, h \le 1,000,000$.

Each of the following *g* lines contain 4 integers, *x*, *y*, *w*, *b*, separated by single spaces. *x* and y represent the coordinates of the bottom left corner of a garden, *w* and *b* represent the *x*-span and *y*-span of the garden. Gardens do not overlap buildings or other gardens. $0 \le x, y \le 1,000,000$ and
$1 \le w, b \le 1,000,000$.

Three zeros, separated by single spaces, on a line by themselves indicate the end of data and should not be processed.

Output

For each test case, the output starts with a line that begins with the word "Test" followed by a space and then an integer *n*. *n* is the number of the test case starting with the value "1".

Each of the following *g* lines begins with an integer, which is a garden number followed, after a blank space, by the assessment of the whole garden area as "safe" or "vulnerable". Gardens are assigned numbers, starting with the value "1", according to the order of their appearance in the input data, and must appear in that order in the output.

Sample Input	Output for the Sample Input
5 3 2 4 10 8 4 2 12 4 4 6 1 4 4 2 5 1 1 8 1 3 1 1 1 1 3 5 1 11 3 3 12 10 4 4 7 3 4 7 1 3 1 1000 2000 1 1 1000 2000 2000 1 1 2001 2000 1 1 2000 1999 1 1 0 0 0	Test 1 1 vulnerable 2 safe 3 safe Test 2 1 safe 2 vulnerable 3 vulnerable