Contents

1 The Dreadful Seven

Super Mario is studying how to use a mysterious Power Square. The Power Square is $n \times n$ with integer values between 0 and $n^2 - 1$.

A number y is a neighbor of another number x in the Power Square if y is directly above or below x, or directly to the left or right of x. Rosalina asks Super Mario to find all the numbers in the Power Square that are neighbors of the number 7, since she can tell that those numbers are quite nervous.

"Why are the numbers scared of seven?" Mario asks Rosalina.

"Because seven ate nine!" Rosalina exclaims.

Input/Output Format:

Input is a description of of the Power Square, followed by a number of commands. The first line is the size of the Power Square n. You may assume $n \leq 100$. The second line contains the n^2 values in the Power Square, separated by spaces. Values start from the top left corner and move from left to right, moving down one row to the leftmost position when a row is filled.

Following the Power Square description are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command "SHOW" causes the current state of the Power Square to be displayed in $n \times n$ form (each row of n values on a single line, separated by spaces), followed by a blank line.
- The command "NEIGHBORS" is followed by a value x in the Power Square. The values neighboring x are output and displayed on a single line (in the order: above, left, right, and below x), separated by spaces. You may assume that x is always in the Power Square.

Examples:

Example 1 Input:	Example 1 Output:
3	8 7 6
8 7 6 5 4 3 2 1 0	5 4 3
SHOW	2 1 0
NEIGHBORS 7	
NEIGHBORS 1	8 6 4
NEIGHBORS 4	4 2 0
	7 5 3 1

Example 2 Input:	Example 2 Output:
4	0 1 2 3
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	4 5 6 7
SHOW	8 9 10 11
NEIGHBORS 7	12 13 14 15
NEIGHBORS 1	
NEIGHBORS 8	3 6 11
NEIGHBORS 14	0 2 5
	4 9 12
	10 13 15

2 Manipulating the Power Square

Rosalina finds Super Mario puzzling over the Power Square and gives him another hint on how to unlock its power. ”Keep swapping 0 with one of its neighbors! I'll tell you which neighbor,” Rosalina says to Mario. Rosalina then gives Mario a sequence of directions on which neighbor to swap with 0.

Input/Output Format:

Input is a description of of the Power Square, followed by a number of commands. The first line is the size of the Power Square n. You may assume $n \leq 100$. The second line contains the n^2 values in the Power Square, separated by spaces. Values start from the top left corner and move from left to right, moving down one row to the leftmost position when a row is filled.

Following the Power Square description are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command ”SHOW” causes the current state of the Power Square to be displayed in $n \times n$ form (each row of n values on a single line, separated by spaces), followed by a blank line.
- The command ”MOVE” is followed by one more more moves: ”up”, ”down”, ”left”, or ”right”. Each move is executed as follows:
  - For ”up”, swap 0 with its neighbor above.
  - For ”left”, swap 0 with its left neighbor.
  - For ”right”, swap 0 with its right neighbor.
  - For ”down”, swap 0 with its neighbor below.

  If move attempts to swap 0 with a non-existent neighbor, then output ”FAILED” on a single line and stop attempting the remaining moves. If the move succeeds, ”MOVED” is output on a single line. In either case, the state of the Power Square is changed to reflect the moves made.

Examples:

Example 1 Input:	Example 1 Output:
3	8 7 6
8 7 6 5 4 3 2 1 0	5 4 3
SHOW	2 1 0
MOVE up	
SHOW	MOVED
	8 7 6
	5 4 0
	2 1 3

Example 2 Input:	Example 2 Output:
3	8 7 6
8 7 6 5 4 3 2 1 0	5 4 3
SHOW	2 1 0
MOVE up right	
	FAILED

3 Evaluating Army Teams

Bowser is forming a Koopa army to invade the Mushroom Kingdom! His army consists of a number of teams of different types of Koopas. After a series of training battles against Bullet Bills and Chain Chomps, Bowser noticed that his teams are more effective if: 1) teams are made up of mostly the same member type, and 2) members of a particular type are on the same team. Ideally the army is most effective when each team is composed of the same member type, and all member types are on the same team.

To evaluate how far his Koopa army is to the ideal team membership, Bowser found he could use the metric *Variation of Information Distance*, or V-score for short. Lower V-scores are preferable since they indicate the solution is closer to the ideal solution. The V-score for a team x is calculated as:

$$v(x) = p_x \times \log(p_x) - 2 \times \sum_d (p_x \times p_d \times \log(p_x \times p_d))$$

where $\sum_d$ is computed for all member types d, and

p_x = fraction of army (excluding commanders) on team x
p_d = fraction of team x with member type d

Note that p_d may be zero if no members of that type are on the team, in which case the multiplier is 0 and the logarithm need not be computed. Your task is to help Bowser calculate V-scores for his teams.

For instance, suppose you want to calculate the V-score for an army with two teams, Team x with 2 Troopas, and Team y with 2 Spinys. Then

$p_x = 0.5$, and $p_{Troopa} = 1.0$ and $p_{Spiny} = 0.0$.

yielding

$$v(x) = 0.5 \times \log(0.5) - 2 \times ((0.5 \times 1.0 \times \log(0.5 \times 1.0)) + (0.5 \times 0.0 \times \log(0.5 \times 0.0)))$$

The result is approximately 0.150515. The V-score for Team y is the same, since the values for p_{Troopa} and p_{Spiny} are simply reversed. In comparison, suppose Team x and Team y had one Troopa and one Spiny each. Then

$p_x = 0.5$, and $p_{Troopa} = 0.5$ and $p_{Spiny} = 0.5$.

yielding

$$v(x) = 0.5 \times \log(0.5) - 2 \times ((0.5 \times 0.5 \times \log(0.5 \times 0.5)) + (0.5 \times 0.5 \times \log(0.5 \times 0.5)))$$

The result is approximately 0.451545, much higher, so we prefer the first set of teams, as expected.

Input/Output Format:

Input is a description of teams, followed by a list of commands.

Team descriptions begin with a line listing different types of team members, separated by spaces. The next line contains the number of team rosters. Team rosters follow, each on a separate line. Each team roster begins with the name of the team commander, followed by team members, all separated by spaces. You may assume that each team has at least one team member.

Following the team rosters are a list of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command "VSCORE" is followed by the name of a team commander x. The output is a single line that begins with x, followed by the V-score for the entire teams supervised by x. V-scores are output as floating point numbers displayed with 6 decimal places.

Examples:

Example 1 Input:	Example 1 Output:
Troopa Paratroopa	Sergeant1 0.150515
2	Sergeant2 0.150515
Sergeant1 Troopa Troopa	
Sergeant2 Paratroopa Paratroopa	
VSCORE Sergeant1	
VSCORE Sergeant2	

Example 2 Input:	Example 2 Output:
Troopa Paratroopa	Sergeant1 0.451545
2	Sergeant2 0.451545
Sergeant1 Troopa Paratroopa	
Sergeant2 Paratroopa Troopa	
VSCORE Sergeant1	
VSCORE Sergeant2	

Example 3 Input:	Example 3 Output:
Troopa Paratroopa Spiny	Sergeant1 0.156863
3	Sergeant2 0.139794
Sergeant1 Troopa Troopa Troopa	Sergeant3 0.150515
Sergeant2 Paratroopa Paratroopa	
Sergeant3 Spiny Spiny Spiny Spiny Spiny	
VSCORE Sergeant1	
VSCORE Sergeant2	
VSCORE Sergeant3	

Example 4 Input:	Example 4 Output:
Troopa Paratroopa Spiny	Sergeant1 0.322724
3	Sergeant2 0.260206
Sergeant1 Troopa Troopa Spiny	Sergeant3 0.563212
Sergeant2 Paratroopa Spiny	
Sergeant3 Spiny Spiny Troopa Paratroopa Spiny	
VSCORE Sergeant1	
VSCORE Sergeant2	
VSCORE Sergeant3	

4 Organizing Bowser's Army

After the latest training battle against the Monty Moles, Bowser noticed his army was getting too large and complex to command on his own. To solve this problem, Bowser decides to promote a number of more experienced team commanders to higher ranks, and have them supervise other team commanders and each other.

Newly promoted commanders no longer directly command teams, instead they supervise two or more commanders. Command assignments are made so that everyone in the army has exactly one direct supervisor. The entire army is ultimately indirectly under the command of a single commander, in a tree-like command hierarchy.

You are tasked with helping Bowser organize his army teams and commanders To help Bowser keep track of his army's command hierarchy, you must help him find the direct supervisor for each commander, and report information on the team members directly and indirectly supervised by each commander.

Input/Output Format:

Input is a list team rosters, followed by a list of command assignments. A list of commands are at the end.

Team descriptions begin with a line listing different types of team members, separated by spaces. The next line contains n, the number of team rosters. You may assume $n \leq 5000$. Team rosters follow, each on a separate line. Each team roster begins with the name of the team commander, followed by team members, all separated by spaces. You may assume that each team has at least one team member.

After team rosters are the command assignments. The first line line contains the number of assignments. Command assignments follow, each on a separate line. Each command assignment begins with the name of a commander x, followed by a list of commanders directly supervised by x.

Following the team rosters are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command "ROSTER" is followed by the name of a commander x.

  The output for the command is a single line beginning with x and listing the number and type of subordinates for any teams directly or indirectly commanded by x. The number of members of a particular type must come before the name of the member type. The member types must be listed in the same order they were listed in the input. All output on the line is separated by a single space.

- The command "COMMANDER" is followed by the name of a team commander x. The output for the command is a single line listing x, the symbol $>$, followed by the direct commander of x. If x has no commander output "NONE" after x instead.

Examples:

Example 1 Input:	Example 1 Output:
Troopa Paratroopa	Captain1 4 Troopa 2 Paratroopa
3	Sergeant1 > Captain1
Sergeant1 Troopa Troopa	Sergeant2 > Captain1
Sergeant2 Troopa Paratroopa	Sergeant3 > Captain1
Sergeant3 Paratroopa Troopa	Captain1 > NONE
1	
Captain1 Sergeant1 Sergeant2 Sergeant3	
ROSTER Captain1	
COMMANDER Sergeant1	
COMMANDER Sergeant2	
COMMANDER Sergeant3	
COMMANDER Captain1	

Example 2 Input:	Example 2 Output:
Troopa Paratroopa Spiny	Bowser 4 Troopa 2 Paratroopa 3 Spiny
4	General3 3 Troopa 0 Paratroopa 1 Spiny
Sergeant4 Troopa Spiny Troopa	Major2 1 Troopa 2 Paratroopa 2 Spiny
Captain3 Paratroopa Spiny Spiny	General3 > Bowser
Sergeant2 Troopa Paratroopa	Major2 > Bowser
Captain1 Troopa	Captain1 > General3
3	
Major2 Sergeant2 Captain3	
General3 Sergeant4 Captain1	
Bowser General3 Major2	
ROSTER Bowser	
ROSTER General3	
ROSTER Major2	
COMMANDER General3	
COMMANDER Major2	
COMMANDER Captain1	

5 Bowser's Command Team

After promoting and appointing a number of team commanders to higher rank, Bowser had a disastrous training battle against the Piranha Plants. Too many commanders were giving conflicting orders to the same team, and the entire army was hopelessly confused.

To solve this problem, you are asked to help Bowser select *command teams*, groups of army commanders such that every team in the army is commanded by *exactly* one commander on the command team. Bowser reasons that by assigning authority to only members of a command team during battle, there will be no conflicting orders issued to the same team.

Two obvious command teams are 1) the top-level commander in charge of the entire army (usually Bowser himself), and 2) the team leader for each team.

Input/Output Format:

Input is a list team rosters, followed by a list of command assignments. A list of commands are at the end.

Team descriptions begin with a line listing different types of team members, separated by spaces. The next line contains n, the number of team rosters. You may assume $n \leq 5000$. Team rosters follow, each on a separate line. Each team roster begins with the name of the team commander, followed by team members, all separated by spaces. You may assume that each team has at least one team member.

After team rosters are the command assignments. The first line line contains the number of assignments. Command assignments follow, each on a separate line. Each command assignment begins with the name of a commander x, followed by a list of commanders directly supervised by x.

Following the team rosters are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command "COMMANDTEAM" is followed by a list of names of commanders separated by spaces.

  The output for the command is a single line containing "YES", if the list of commanders forms a command team, and "NO" otherwise.

Examples:

Example 1 Input:	Example 1 Output:
Troopa Paratroopa	YES
3	NO
Sergeant1 Troopa Troopa	YES
Sergeant2 Troopa Paratroopa	NO
Sergeant3 Paratroopa Troopa	
1	
Captain1 Sergeant1 Sergeant2 Sergeant3	
COMMANDTEAM Captain1	
COMMANDTEAM Sergeant1 Sergeant2	
COMMANDTEAM Sergeant2 Sergeant1 Sergeant3	
COMMANDTEAM Captain1 Sergeant3	

Example 2 Input:	Example 2 Output:
Troopa Paratroopa Spiny	YES
4	YES
Sergeant4 Troopa Spiny Troopa	YES
Captain3 Paratroopa Spiny Spiny	YES
Sergeant2 Troopa Paratroopa	NO
Captain1 Troopa	NO
3	NO
Major2 Sergeant2 Captain3	
General3 Sergeant4 Captain1	
Bowser General3 Major2	
COMMANDTEAM Bowser	
COMMANDTEAM Major2 General3	
COMMANDTEAM Captain3 Sergeant2 General3	
COMMANDTEAM Sergeant4 Major2 Captain1	
COMMANDTEAM Bowser Captain1	
COMMANDTEAM General3 Major2 General3	
COMMANDTEAM Major2 Captain1	

6 The Composite Galaxy

Super Mario takes a break from exploring the Power Square to do some exploring. He finds himself in a galaxy where, in order to earn power stars, he needs to find consecutive composite numbers. Composite numbers are numbers with more than two positive integer divisors. For example, the sequence $90 = 2 \times 45$, $91 = 7 \times 13$, $92 = 2 \times 46$, $93 = 3 \times 31$, $94 = 2 \times 47$, $95 = 5 \times 19$, $96 = 2 \times 48$ is a list of 7 consecutive composite numbers. Note that we also presented evidence that they are composite (since 1 and the original number are always positive integer divisors, a third positive integer divisor is all that's needed).

Write a program to help Super Mario!

Input/Output Format:

The input is a list of integer ranges for which you must find the longest consecutive sequence of composite numbers. Each line of input provides a single range, consisting of two positive integers, the lower bound and the upper bound (both inclusive), separated by a space.

You may assume that all numbers in the range are less than Integer.MAX_VALUE.

The output is a list of consecutive sequences of composite numbers found for each range. Each line of output provides a consecutive sequence of composite numbers, consisting of the smallest and largest values within the sequence, separated by a space.

If there are multiple longest consecutive sequences, output the sequence with larger values.

Your program must be able to compute the answer in under 10 seconds, even for ranges of integers with values over 10 million. Your program must also not run out of memory.

Examples:

Example 1 Input:	Example 1 Output:
10 20	14 16
20 40	32 36

Example 2 Input:	Example 2 Output:
90 96	90 96
2 1000	888 906
10000 20000	19610 19660

Example 3 Input:	Example 3 Output:
987654 1234567	1098848 1098952

Hint: Don't try to solve the problem in a brute-force manner by checking for divisors for all numbers in the range, since it will take too long for larger numbers and ranges.

7 Find the Best Command Team!

Using command teams works well and allows Bowser to organize and command a large army of Koopas. After one last training battle against Thwomps and Wigglers, Bowser is ready to launch his invasion of the Mushroom Kingdom and claim victory over Super Mario and Princess Peach.

The last task Bowser would like to accomplish is to combine his insights for V-scores and command teams, by selecting a command team with the best (i.e., minimal) V-score to organize his army.

The V-score for a command team is simply the sum of the V-scores for the individual teams supervised directly or indirectly by each commander on the command team. When calculating the V-score for a commander x supervising other commanders, simply treat the members of all teams commanded by x as belonging to a single large team (as you did for the ROSTER command in Problem 4). For instance, the V-score of the overall army commander can be calculated as if the entire army is in a single team.

Unfortunately the number of possible command teams is astronomical for a large number (e.g., $>$ 500) of teams, so it's not feasible to simply calculate the V-score for every possible command team and select the team with the lowest V-score. Nonetheless, you must help Bowser succeed!

Your solution must be able to handle 500 teams in under 30 seconds.

Input/Output Format:

Input is a list team rosters, followed by a list of command assignments. A list of commands are at the end.

Team descriptions begin with a line listing different types of team members, separated by spaces. The next line contains n, the number of team rosters. You may assume $n \leq 5000$. Team rosters follow, each on a separate line. Each team roster begins with the name of the team commander, followed by team members, all separated by spaces. You may assume that each team has at least one team member.

After team rosters are the command assignments. The first line line contains the number of assignments. Command assignments follow, each on a separate line. Each command assignment begins with the name of a commander x, followed by a list of commanders directly supervised by x.

Following the team rosters are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command "VSCORE" is followed by the name of a team commander x. The output is a single line that begins with x, followed by the V-score for the entire team supervised by x.
- The command "BESTCOMMAND" has no parameters. The output for the command is the lowest V-score of all command teams, followed by the list of commanders in the command team in lexicographically sorted order (you may use Arrays.sort).

V-scores are output as floating point numbers displayed with 6 decimal places.

Examples:

Example 1 Input:	Example 1 Output:
Troopa Paratroopa	Sergeant1 0.150515
2	Sergeant2 0.150515
Sergeant1 Troopa Troopa	Captain1 0.602060
Sergeant2 Paratroopa Paratroopa	0.301030 Sergeant1 Sergeant2
1	
Captain1 Sergeant1 Sergeant2	
VSCORE Sergeant1	
VSCORE Sergeant2	
VSCORE Captain1	
BESTCOMMAND	

Example 2 Input:	Example 2 Output:
Troopa Paratroopa	Sergeant1 0.451545
2	Sergeant2 0.451545
Sergeant1 Troopa Paratroopa	Captain1 0.602060
Sergeant2 Paratroopa Troopa	0.602060 Captain1
1	
Captain1 Sergeant1 Sergeant2	
VSCORE Sergeant1	
VSCORE Sergeant2	
VSCORE Captain1	
BESTCOMMAND	

Example 3 Input:	Example 3 Output:
Troopa Paratroopa Spiny	Sergeant1 0.156864
3	Sergeant2 0.139794
Sergeant1 Troopa Troopa Troopa	Sergeant3 0.150515
Sergeant2 Paratroopa Paratroopa	Captain1 0.894345
Sergeant3 Spiny Spiny Spiny Spiny Spiny	0.447173 Sergeant1 Sergeant2 Sergeant3
1	
Captain1 Sergeant1 Sergeant2 Sergeant3	
VSCORE Sergeant1	
VSCORE Sergeant2	
VSCORE Sergeant3	
VSCORE Captain1	
BESTCOMMAND	

Example 4 Input:	Example 4 Output:
Troopa Paratroopa Spiny	Sergeant1 0.322724
3	Sergeant2 0.260206
Sergeant1 Troopa Troopa Spiny	Sergeant3 0.563212
Sergeant2 Paratroopa Spiny	Captain1 0.894345
Sergeant3 Spiny Spiny Troopa Paratroopa Spiny	0.894345 Captain1
1	
Captain1 Sergeant1 Sergeant2 Sergeant3	
VSCORE Sergeant1	
VSCORE Sergeant2	
VSCORE Sergeant3	
VSCORE Captain1	
BESTCOMMAND	

8 Solve the Power Square?

Rosalina summons Mario and tell him Bowser will soon be invading the Mushroom Kingdom.

"But how do I stop his army?" Mario asks.

"Solve the Power Square to get the Super Power Star!" Rosalina tells Mario.

Your goal is to help Mario solve the Power Square. Given a Power Square, you must find the shortest sequence of moves which will sort all the values in the Power Square. The only moves that you are allowed to make are those that swap 0 with one of its neighbors.

Input/Output Format:

Input is a description of of the Power Square, followed by a number of commands. The first line is the size of the Power Square n. You may assume $n \leq 100$. The second line contains the n^2 values in the Power Square, separated by spaces. Values start from the top left corner and move from left to right, moving down one row to the leftmost position when a row is filled.

Following the Power Square description are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command "SHOW" causes the current state of the Power Square to be displayed in $n \times n$ form (each row of n values on a single line, separated by spaces), followed by a blank line.
- The command "SOLVE" is followed by x, the number of moves you are allowed to use to solve the Power Square.

  If there is no sequence of moves of length x or less that can solve the Power Square, output "FAILED" on a single line.

  Otherwise find the shortest sequence of moves that can solve the Power Square, and output it as "MOVE" followed by the moves ("up", "left", "right", "down") on a single line. If there are multiple shortest sequences, you may output any solution.

Your solver must be efficient enough to solve a 4×4 puzzle of 12 moves in under 30 seconds.

Examples:

Example 1 Input:	Example 1 Output:
3	1 2 0
1 2 0 3 4 5 6 7 8	3 4 5
SHOW	6 7 8
SOLVE 1	
SOLVE 2	FAILED
SOLVE 3	MOVE left left
	MOVE left left

Example 2 Input:	Example 2 Output:
4	4 1 2 3
4 1 2 3 5 9 6 7 8 0 10 11 12 13 14 15	5 9 6 7
SHOW	8 0 10 11
SOLVE 2	12 13 14 15
SOLVE 3	
	FAILED
	MOVE up left up

Practice 1 Sorted Power Square

Super Mario finds a mysterious a $n \times n$ Power Square with integer values between 0 and $n^2 - 1$. Rosalina asks Super Mario whether the values in the Power Square are sorted from smallest to largest value.

”Why do you want to know whether the values are sorted?” Mario asks.

”I’ll tell you this afternoon,” Rosalina replies.

Input/Output Format:

Input is a description of of the Power Square, followed by a number of commands. The first line is the size of the Power Square n. The second line contains the n^2 values in the Power Square, separated by spaces. Values start from the top left corner and move from left to right, moving down one row to the leftmost position when a row is filled.

Following the Power Square description are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command ”SHOW” causes the current state of the Power Square to be displayed in $n \times n$ form (each row of n values on a single line, separated by spaces), followed by a blank line.
- The command ”CHECK” analyzes the current state of the Power Square to see whether its values are in sorted order, from smallest value to largest. The result is output on a single line as ”SORTED” if the Power Square is sorted, and as ”NOT SORTED” if the Power Square is not sorted.

Examples:

Example 1 Input:	Example 1 Output:
2	0 1
0 1 2 3	2 3
SHOW	
CHECK	SORTED

Example 2 Input:	Example 2 Output:
3	1 2 3
1 2 3 4 5 6 7 8 0	4 5 6
SHOW	7 8 0
CHECK	
	NOT SORTED

Practice 2 Power Square Swaps

Super Mario finds a mysterious a $n \times n$ Power Square with integer values between 0 and $n^2 - 1$. Rosalina teaches Super Mario how to swap two values with each other in the square with great concentration.

Input/Output Format:

Input is a description of of the Power Square, followed by a number of commands. The first line is the size of the Power Square n. The second line contains the n^2 values in the Power Square, separated by spaces. Values start from the top left corner and move from left to right, moving down one row to the leftmost position when a row is filled.

Following the Power Square description are a number of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command "SHOW" causes the current state of the Power Square to be displayed in $n \times n$ form (each row of n values on a single line, separated by spaces), followed by a blank line.
- The command "SWAP" is followed by the two values in the Power Square that need to be swapped. Each swap is applied to the result of the previous swap query. No output is produced.

Examples:

Example 1 Input:	Example 1 Output:
2	0 1
0 1 2 3	2 3
SHOW	
SWAP 1 2	0 2
SHOW	1 3
SWAP 1 3	
SHOW	0 2
	3 1

Example 2 Input:	Example 2 Output:
3	1 2 3
1 2 3 4 5 6 7 8 0	4 5 6
SHOW	7 8 0
SWAP 1 0	
SWAP 8 2	0 8 3
SHOW	4 5 6
	7 2 1

Practice 3 Team Rosters

Bowser is forming a Koopa army to invade the Mushroom Kingdom! His army consists of a number of teams.

Input/Output Format:

Input is a description of teams, followed by a list of commands.

Team descriptions begin with a line listing different types of team members, separated by spaces. The next line contains the number of team rosters. Team rosters follow, each on a separate line. Each team roster begins with the name of the team commander, followed by team members, all separated by spaces. You may assume that each team has at least one team member.

Following the team rosters are a list of commands, with each command on a separate line. Each command begins with the name of the command, followed by any additional command parameters.

- The command ”ROSTER” is followed by the name of a team commander x or ”ALL”.

  The output for the command for x is a single line that begins with x and lists the number and type of subordinates for the team commanded by x. The output for ”ALL” begins with ”ALL” and lists the number and type of members for the entire army. The number of members of a particular type must come before the name of the member type. The member types must be listed in the same order they were listed in the input. All output on the line is separated by a single space.

Examples:

Example 1 Input:	Example 1 Output:
Troopa Paratroopa	Sergeant1 2 Troopa 0 Paratroopa
3	Sergeant2 1 Troopa 1 Paratroopa
Sergeant1 Troopa Troopa	Sergeant3 1 Troopa 1 Paratroopa
Sergeant2 Troopa Paratroopa	ALL 4 Troopa 2 Paratroopa
Sergeant3 Paratroopa Troopa	
ROSTER Sergeant1	
ROSTER Sergeant2	
ROSTER Sergeant3	
ROSTER ALL	

Example 2 Input:	Example 2 Output:
Troopa Paratroopa Spiny	Captain1 1 Troopa 0 Paratroopa 0 Spiny
4	Sergeant2 1 Troopa 1 Paratroopa 0 Spiny
Sergeant4 Troopa Spiny Troopa	Captain3 0 Troopa 1 Paratroopa 2 Spiny
Captain3 Paratroopa Spiny Spiny	ALL 4 Troopa 2 Paratroopa 3 Spiny
Sergeant2 Troopa Paratroopa	
Captain1 Troopa	
ROSTER Captain1	
ROSTER Sergeant2	
ROSTER Captain3	
ROSTER ALL	