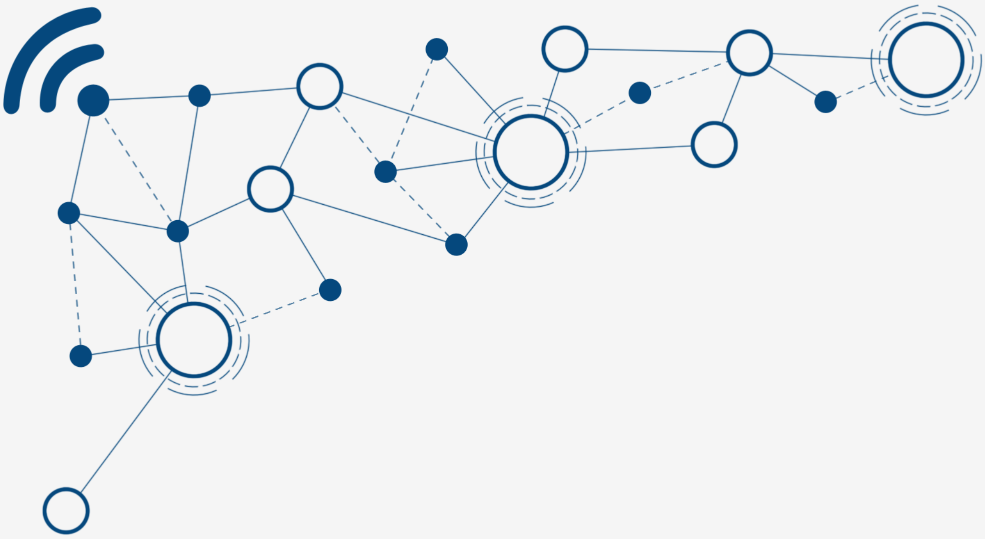

National Olympiad in Informatics

Finals Round 1

Important! Read the following:

Hidden Test Cases. Your solution will be checked by running it against one or more (usually several) hidden test cases. You will not have access to these cases, but a correct solution is expected to handle them correctly.

Strict Output Format. The output checker is **strict**. Follow these guidelines strictly:

- It is **space sensitive**. Do not output extra leading or trailing spaces. Do not output extra blank lines unless explicitly stated.
- It is **case sensitive**. So, for example, if the problem asks for the output in lowercase, follow it.
- Do not print any tabs. (No tabs will be required in the output.)
- Do not output anything else aside from what's asked for in the Output section. So, do not print things like "`Please enter t`".

Not following the output format strictly and exactly will likely result in the verdict "*Output isn't correct*".

Use Standard I/O. Do not read from, or write to, a file. You must read from the standard input and write to the standard output.

Submit Code Only. Only include **one** file when submitting: the source code (.cpp, .py, etc.) and nothing else.

No Java Package. For Java submissions, do not include a `package` line.

No Weird Filenames. Only use letters, digits and underscores in your filename. Do not use spaces or other special symbols.

Use Fast I/O. Many problems have large input file sizes, so use fast I/O. For example:

- In C/C++, use `scanf` and `printf`.
- In Python, use `sys.stdin.readline()`

Flush On Interactive Problems. On interactive problems, make sure to **flush** your output stream after printing.

- In C++, use `fflush(stdout);` or `cout << endl;`
- In Python, use `sys.stdout.flush()` or `print(flush=True)`
- For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the contest! 😃

National Olympiad in Informatics

FINALS 1

Contents

A: Method of Least Squares 3

B: Astig Runnings 6

C: Electra Boom 9

D: The Manga Guide to Algorithms 15

Notes

- Many problems have large input file sizes, so use fast I/O. For example:
  - In C/C++, use `scanf` and `printf`.
  - In Python, use `sys.stdin.readline()`
- On interactive problems, make sure to **flush** your output stream after printing.
  - In C++, use `fflush(stdout);` or `cout << endl;`
  - In Python, use `sys.stdout.flush()` or `print(flush=True)`
  - For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the problems!

Problem D
The Manga Guide to Algorithms

After many years, NOI.PH has finally released another entry in the Manga Guide series by Saji Tan—this is The Manga Guide to Algorithms! We hope that this and our many other outreach efforts help competitive programming reach an ever wider audience.

The scene has grown so much over the years, and we hope that it will continue to grow and prosper in the years to come.

To celebrate, NOI.PH and Saji Tan have decided to organize a book-launching event for the new Manga Guide, with free food afterwards for all invited guests (catered by a celebrity chef)! Help us decide the menu, taking into account people's preferences, but also keeping in mind any allergies.

There are n guests attending the event, labeled 1 through n. There are 10^9 possible types of ingredients in the world that our celebrity chef can use, labeled 1 through 10^9. A dish is assembled by choosing some **non-empty** subset of these ingredients to put in that dish.

Prior to the event, we collected m *opinions* from the guests. Each opinion is in one of two forms:

- "Guest i **must have** ingredient t"; or
- "Guest i **cannot eat** ingredient t".

To keep the menu simple, our celebrity chef will only put **two** dishes on the menu. Please determine if it is possible to make it such that each person has a dish that they would love to eat (that is, each person's meal should have all the ingredients they must have, but none of the ingredients they cannot eat).

Also, we will be preparing packed meals for the event, so we would have to determine ahead of time which dish to make for each guest.

If yes, please also tell us exactly what to do! That, is, output three things:

- Which ingredients to put in the first dish.
- Which ingredients to put in the second dish.
- An assignment of, for each of the n guests, which of the two dishes will be made for them.

Among all possible assignments, it is also preferable for the number of guests eating each dish to be *as balanced as possible*. That is, let A be the number of guests assigned to eat the first dish, and let B be the number of guests assigned to eat the second dish. We would *prefer* if $|A - B|$ is minimized. If not, but the answer would otherwise be valid, then you can still earn partial points.

Input Format

The first line of input contains two space-separated integers n and m.

Then, m lines follow, each describing an opinion. Each line consists of three space-separated tokens, each one either:

- "i `:)` t" or
- "i `>:(` t"

meaning that guest i loves (or hates) ingredient t, respectively.

Output Format

First, output a line containing the token `:)` or `>:(`, depending on whether the task is possible or not (respectively).

If `:)`, please also output the following, each in their own line:

- First, a positive integer k_A, the number of ingredients to put in the first dish.
- Then, k_A distinct space-separated integers, each from 1 to 10^9—the ingredients to put in the first dish.

- Then, a positive integer k_B, the number of ingredients to put in the second dish.
- Then, k_B distinct space-separated integers, each from 1 to 10^9—the ingredients to put in the second dish.
- Finally, n space-separated integers, each one either 1 or 2. The ith integer represents which meal should be eaten by the ith guest (1 for the first dish, and 2 for the second dish).

Each of k_A and k_B should be at least 1 and at most 4×10^5. If there are multiple possible solutions, any will be accepted (and scored accordingly).

Constraints

For all subtasks
$2 \leq n \leq 2 \times 10^5$ $0 \leq m \leq 3 \times 10^5$ $1 \leq$ each ingredient number $\leq 10^9$ No guest says their opinion about the same ingredient twice No guest both likes and dislikes the same ingredient.

Subtask	Points	Constraints
1	**15**	$n \leq 10$ $m \leq 2000$
2	**35**	$n \leq 5000$
3	**30**	Each ingredient is mentioned in no more than two opinions.
4	**20**	No further constraints.

There is also partial scoring:

- You get 0 points for this subtask if there exists a test file under this subtask where either of the following holds:
  - Your `:)` or `>:(` answer was incorrect.
  - For some `:)` solution, your construction was invalid or incorrect.
- If not, you get 60% of the points for this subtask if there exists a test file under this subtask such that: your construction was valid, but did not minimize $|A - B|$.
- Otherwise, you get 100% of the points for this subtask.

Sample I/O

Input 1	Output 1
8 9 1 :) 123 2 :) 123 3 :) 123 4 >:(123 4 >:(456 5 >:(456 6 :) 456 6 >:(789 7 :) 789	:) 3 123 456 1000000000 4 789 404 420 1000000000 1 1 1 2 2 1 2 2

Input 2	Output 2
3 6 1 :) 997 2 >:(997 1 >:(998 2 :) 998 3 :) 997 3 :) 998	>:(