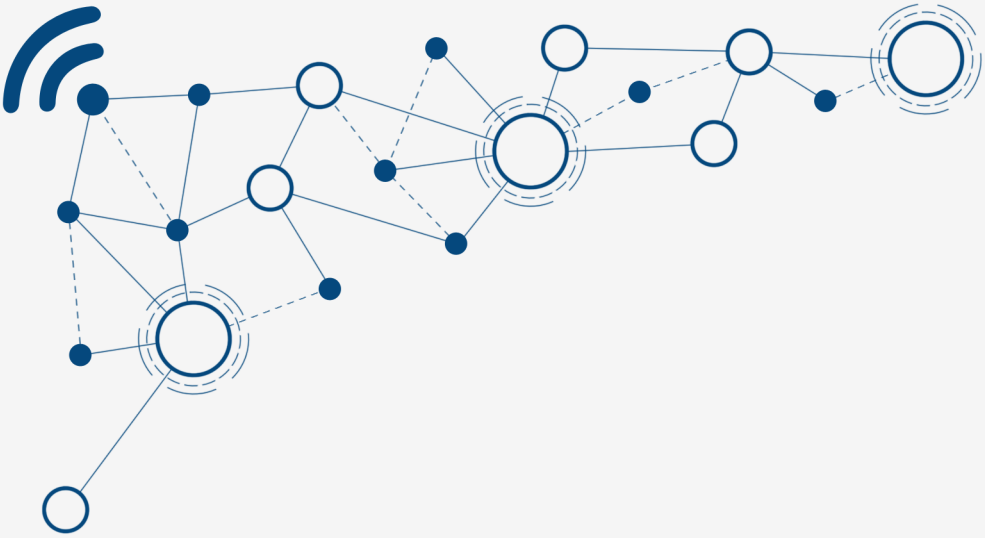
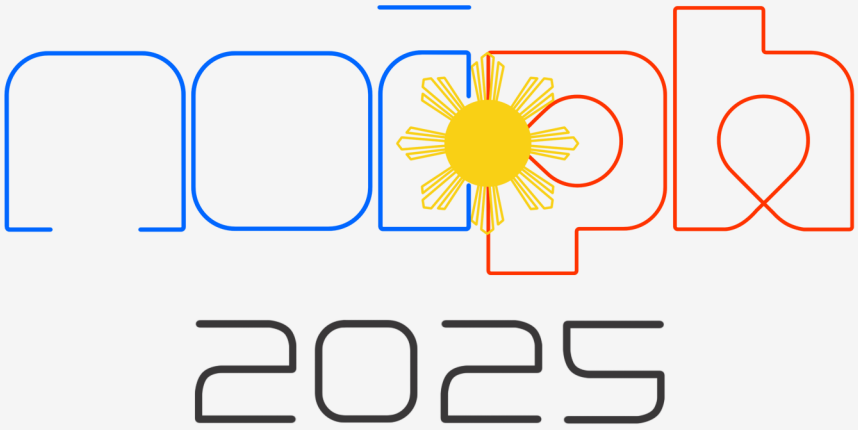
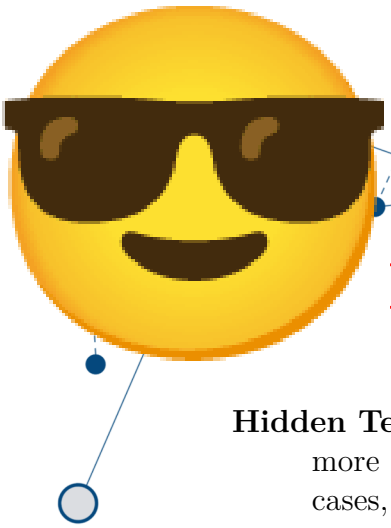




National Olympiad in Informatics

Finals Round 2



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! 😃

Contents

A: Incompetent Interactor 3

B: The Astounding Race 6

C: Fast Threerier Transform 9

D: CJ's Marble Runs 13

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 A
Incompetent Interactor

Alice has a great idea for an interactive problem for the NOI.PH 2025 Finals!

It goes like this:

Alice is thinking of a secret integer A, from 1 to m (inclusive). Bob's job is to guess what it is.

Bob can ask her n questions, each of the following form:

- "`Is A {cmp} {x} ?`" (without the quotes)

where ***`x`*** *is some positive integer, and* ***`cmp`*** *is one of*

<	<=	==
>	>=	!=

with their usual familiar meanings. Alice will respond "Yes" or "No" to each one.

For example, suppose $m = 10$, and Bob asks the $n = 2$ questions

- `Is A > 6 ?`
- `Is A < 9 ?`

If Alice had chosen $A = 7$, then she should answer "Yes" and then "Yes".

Alice and Bob try playing this game for real. Bob proceeds to ask Alice his n questions. The issue is that Alice had forgotten her initial number A. Oops! So, she just answers Bob's questions randomly. Bad Alice!

Oh well, as long as her answers are consistent with *some* integer from 1 to m, Bob can't call her out on it. If she ends up giving contradictory answers, though, Bob will be furious!

There are 2^n ways that Alice can choose to answer Bob's n Yes/No questions—call one such way a *response set*.

The **narrowness** of a response set is equal to the number of integers from 1 to m such that if that were the value of A, then this response set would be how Alice should have answered Bob's questions. Then, a response set is called **valid** if its narrowness is nonzero.

In the example from earlier, the narrowness of the response set (Yes, Yes) is 2, since Alice should answer "Yes" and then "Yes" if she had chosen $A = 7$, and also if she had chosen $A = 8$. Since $2 \neq 0$, this response set is valid.

Please answer these two questions:

- How many of these 2^n response sets are valid?
- What is the *sum* of the narrownesses across all 2^n response sets?

Output each answer modulo $10^9 + 7$.

Input Format

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

Then, n lines follow, the ith of which contains Bob's ith question, in the format described above.

Output Format

Output two space-separated integers: the number of valid response sets, and the sum of the narrownesses across all response sets (each modulo $10^9 + 7$).

Constraints

For all subtasks
$1 \leq n \leq 10^5$ $1 \leq m \leq 10^9$ $1 \leq x \leq 10^9$ in each of Bob's questions.

Subtask	Points	Constraints
1	**11**	All questions use `<`
2	**17**	There are no `==` or `!=` questions.
3	**13**	$m \leq 100$ $n \leq 15$
4	**19**	$m \leq 2 \times 10^5$ $n \leq 15$
5	**19**	$n \leq 15$
6	**21**	No further constraints.

Sample I/O

Input 1	Output 1
4 10 Is A <= 7 ? Is A >= 2 ? Is A < 13 ? Is A != 5 ?	4 10

Input 2	Output 2
6 144 Is A <= 1 ? Is A >= 2 ? Is A < 3 ? Is A > 5 ? Is A == 8 ? Is A != 13 ?	6 144

Input 3	Output 3
5 12 Is A == 1 ? Is A == 2 ? Is A == 4 ? Is A == 8 ? Is A == 16 ?	5 12