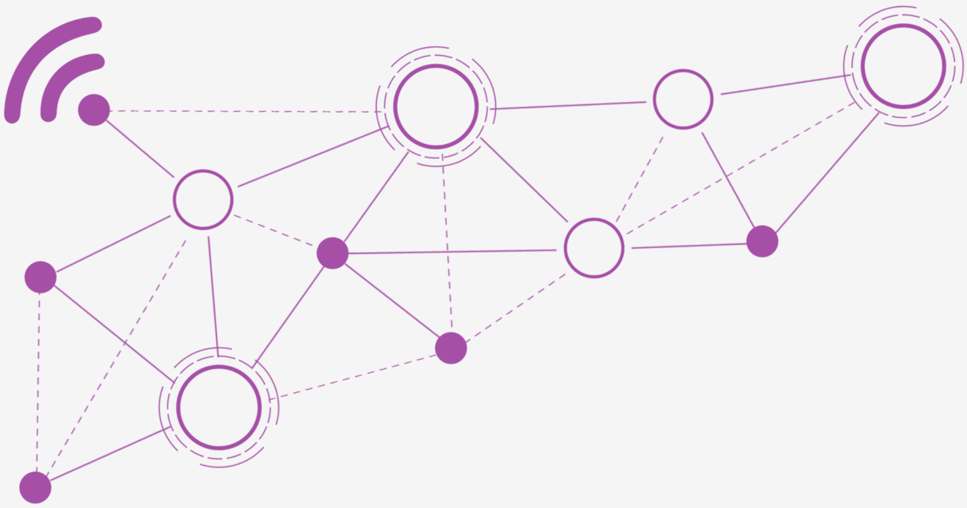


National Olympiad in Informatics

Final Round

Contents

Notes

- Many problems have large input file sizes, so use fast I/O. For example:
  - In C/C++, use `scanf` and `printf`.
  - In Python, `import sys` and then use `sys.stdin.readline()`
- Good luck and enjoy the problems!

Important! Read the following:

- 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.
- 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*".
- Do not read from, or write to, a file. You must read from the standard input and write to the standard output.
- Only include one file when submitting: the source code (.cpp, .py, etc.) and nothing else.
- Only use letters, digits and underscores in your filename. Do not use spaces or other special symbols.
- Many problems have large input file sizes, so use fast I/O. For example:
  - In C/C++, use `scanf` and `printf`.
  - In Python, `import sys` and then use `sys.stdin.readline()`

  We recommend learning and using these functions during the Practice Session.
- Good luck and enjoy the contest!

Problem A
...Are the Bestagons!

Time Limit: 2 seconds

*What is **objectively** the best shape?*

You have six tries, and the first five don't count.

You have somehow found yourself dragged into a hexagon-loving cult. The cult has n members, indexed from 1 to n (because who needs a name and an identity, when you have hexagons?). You are aware of m friendships within the cult; each friendship involves two people u and v, and says that u and v are friends with each other (this is a mutual relationship).

Hexagons are a source of immense power. A *hexagon* is a set of any six **different** people a, b, c, d, e, f, such that all of the following hold:

- a and b are friends
- b and c are friends
- c and d are friends
- d and e are friends
- e and f are friends
- f and a are friends

There may or may not be other friendships among these chosen six, but it doesn't matter. So long as *these* six friendships exist, then the six form a *hexagon*!

So tell me... does this cult contain a hexagon?

(No, there's no subplot about escaping. I think you're just stuck here now.)

Input Format

The first line of input contains two space-separated integers n and m, the number of cult members and the number of friendships.

Then, m lines of input follow, each describing some friendship. Each line contains two space-separated integers u and v, specifying that u and v are friends with one another.

Output Format

Output `YES` if the cult contains a hexagon. Otherwise, output `NO`

Constraints and Subtasks

For all subtasks
$0 \le n, m \le 250$ $1 \le u, v \le n$ No friendship is mentioned more than once in the input. (This includes the fact that if `u v` appears, then `v u` will not.) A person is never described to be friends with themself.

Subtask	Points	Constraints
1	**15**	$m, n \le 6$
2	**12**	$n \le 15$
3	**25**	At most one person has more than 2 friends.
4	**12**	$n \le 70$
5	**36**	No further constraints.

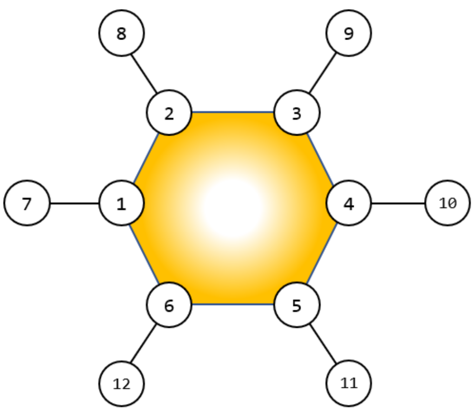
Sample I/O

Input 1	Output 1
12 12 1 2 2 3 3 4 4 5 5 6 1 6 7 1 8 2 9 3 10 4 11 5 12 6	YES

Input 2	Output 2
6 6 1 2 2 3 3 4 4 5 5 1 1 6	NO

Explanation

The following diagram showcases the first sample input. The hexagon has been highlighted; we can take $a = 1$, $b = 2$, $c = 3$, $d = 4$, $e = 5$, and $f = 6$.



Problem B
Intergalactic Word Game

Time Limit: 2 seconds

This is an interactive problem.

The year is 2048. Cleverbot, a digitized version of Cleverlyn's consciousness, has now created Tinikling 10^{100} *using millions of iterations. The entirety of the Philippines is converted into sapin-sapin by a general artificial intelligence. And humanity receives communication from what might be other life, that reads as follows . . .*

Greetings! We represent the Galactic Empire's First Contact Unit, and we are pleased to inform you that your species has developed a suitable level of intelligence to join the Empire, as detailed in the attached reports. As a sign of goodwill, we will offer you a mathematical formula that will shed light on the necessary technological advancements to assimilate into the Empire. We trust that humanity is a very peaceful species, and that Earth is a mostly harmless planet; thus, we expect that this formula will only be used for the good of all. We have sent a probe to orbit your planet to conduct further communications. Good luck!

After studying the schematics given by the aliens, the world's scientists have devised the WRDLS (Worldwide Rebus Derivation Logger System) to send signals to and receive responses from the artificial satellite that is now occupying low Earth orbit. Using the WRDLS, they aim to discover a secret formula F that is composed of n uppercase English letters.

The WRDLS is capable of transmitting n bytes of data at a time to the probe, which is just enough for a string of n characters, which will be denoted as S, and whose ith character ($1 \leq i \leq n$) will be denoted by S_i. After the probe receives the data, it converts the string into a sequence of n **colors**, each of which is either green, yellow or black. The ith color's meaning is as follows.

- Green means that $S_i = F_i$.
- Yellow means that S_i appears somewhere in F, but not at position i.
- Black means that S_i appears nowhere in F.

Then, the alien manning the probe converts this sequence of colors into a sequence of n characters, each of which are either `G`, `Y`, or `B`. If this alien were a human, we could assume that green gets converted to `G`, yellow to `Y`, and black to `B`.

But, for all we know, the language the alien speaks has the words 'ylopa', 'bujt', and 'gaaromo' for what we know as green, yellow, and black, respectively, so they may convert green to `Y`, yellow to `B`, and black to `G`. Furthermore, different aliens may be manning the probe at different times, so the conversion may not be consistent across transmissions!

It is at least known that two characters *within the same transmission* will have the same response character if and only if they had the same response color.

To reiterate: the mapping of colors to characters **may not** follow the human standard, and **may not** even be consistent across transmissions, but the mapping within a particular transmission is always consistent.

Since using the WRDLS takes up a lot of power, we wish to use only a small number of transmissions to obtain the equation. Specifically, we only have enough resources to make T transmissions. Discover the hidden equation, and help establish our dominance among the stars!

Interaction

The first line of input contains an integer n, the length of the hidden word. After receiving it, you must interact with the judge.

To use the WRDLS, output a line containing "`query <S>`", where `<S>` is a string of uppercase English characters, denoting a guess of the equation.

Your program will then receive a string of n characters, each of which is one of `G`, `Y`, or `B`. Consult the Constraints and Subtasks section for their meanings.

Once you have determined the formula, output "`answer <S>`", where `<S>` is what you have determined the formula to be. Your answer will be accepted if, for any possible query, and any possible response to that query, the response would also be valid if the formula was equal to your answer. Note that subtasks are *not* taken into account when determining the validity of an answer to a query.

If at any point you make an invalid query, or if you make more than T queries, your program will receive a single line containing "`ERROR`", after which the judge will immediately quit. Make sure to quit as well, so as to receive the correct verdict.

Caution: Remember to flush your output buffer after every line of output you make.

- C++ users may use `fflush(stdout)` or `cout << flush`.
- Python users may use `sys.stdout.flush()`.

Constraints and Subtasks

For all subtasks
$3 \leq n \leq 10^5$ $T = 35$

If, for any subtask, you make an invalid query, or you use more than T queries, you will receive 0 points. Otherwise, you will receive the number of points as shown.

Subtask	Points	Constraints
1	**20**	The judge will map green, yellow, and black to `G`, `Y`, and `B`, respectively.
2	**20**	The judge will keep a *consistent* mapping of colors to characters across all transmissions.
3	**60**	No further constraints.

Sample I/O

An interaction where the secret formula is `TWOPLUSTWOISFOUR` is shown below.

Spacing is only shown for clarity. Your program must not print any extra blank lines.

Also, note that this interaction is only applicable to subtask 3.

Input	Output
16 BBBYBYGBBYBYBYBB GGBYGYBGGGYBYGGY	query EEQUALSMCSQUARED query ADOTBISABCOSINET answer TWOPLUSTWOISFOUR

Explanation

The probe's first response mapped green, yellow, and black to `G`, `Y`, and `B`, respectively.

The probe's second response mapped green, yellow, and black to `B`, '`Y`, and `G`, respectively.

Since the formula is indeed `TWOPLUSTWOISFOUR`, and you used $3 \leq 35$ queries, your answer would be marked as correct. If there existed some string S such that for any query (not just the ones your program makes), and for any response to that query, that response is also valid if S were the formula, then you would also be marked as correct if you answered that S instead.

Testing Tool

A testing tool written in Python 3 is provided in the attachments in CMS to help with local testing. The testing tool reads input from a file named "`input.txt`", which must contain a single line. This line must contain the formula to guess, and the subtask number, separated by a space. A sample `input.txt` file is also included as an attachment.

To run the testing tool, in your terminal, run

```
python3 interactive_runner.py python3 testing_tool.py
    input.txt -- [COMMAND]
```

where `python3` can be replaced by the command you use to invoke Python 3 in your terminal, and `[COMMAND]` is how you invoke your solution.

If your solution would be marked as Wrong Answer for reasons other than an incorrect final answer, the testing tool will output an appropriate message. Otherwise, it will output a message containing the judge's formula, and your final answer. Note that it does not verify if this answer will be accepted.

Problem C
The Fibonacci Five-Tuplets
Time Limit: 10 seconds

Have you met the Pisano sisters? Quite peculiarly, the five of them are identical five-tuplets! Let's introduce them, from eldest to youngest:

- *Isabella, the reliable older sister,*
- *Dalawahlia, the most direct and outspoken sister,*
- *Tresha, the shy and reclusive sister,*
- *Cuatrona, the bubbly and athletic sister,*
- *Limahong, the reincarnation of the same-named Chinese pirate and warlord who terrorized the Spanish-ruled Northern Philippines in the late 1500s.*

(They usually just call the fifth sister, "Li".)

Simoy Sehs is the Rank 1 Honors Student in his school, and the Pisanos' father had hired him as the sisters' tutor in order to raise their grades. As their tutor, he's supposed to spend time with all five of them equally... and normally the five girls share all their things equally... but recently, Li has started to feel that...

If you love someone, isn't it natural to want them all to yourself?

Simoy's birthday is coming up soon, and the Pisano sisters agreed that they would each get him a Fibonacci number as a gift! The Fibonacci numbers F_n are the elements of the sequence with $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$.

The local temple has n different amulets for sale, indexed from 1 to n, and each one containing a Fibonacci number. Given an array $a_1, a_2, a_3, \ldots, a_n$, the number on the ith amulet is F_{a_i}. This array a may contain duplicates.

Four of the sisters agree on two integers L and R. They promise each other to only buy amulets whose indices lie between L and R inclusive, and they also will respect their birth order with the indices of the amulets they buy. Formally, they can choose any four indices h, i, j, k such that $L \leq h < i < j < k \leq R$, and they will purchase the amulets with those indices.

Li realizes that this is her chance to stand out from her sisters. She arranges to acquire—completely separately—an amulet that contains the number F_x. If the value of her amulet is greater than or equal to the values of *all* her sisters' amulets combined, then surely Simoy will start to see her as someone special!

But Li wonders whether this is even something that is likely to happen, or if it's just a fantasy. Specifically and formally, she needs you to help her answer the following question: Given the values of L, R, and x, **count** the number of four-tuples h, i, j, k such that both of the following hold:

- $L \leq h < i < j < k \leq R$
- $F_{a_h} + F_{a_i} + F_{a_j} + F_{a_k} \leq F_x$

Also, none of the plans are really set in stone, so you'll need to answer q different queries, each with their own L, R, and x. Good luck!

Input Format

The first line of input contains two space-separated integers n and q, the number of different amulets sold at the temple, and the number of different queries you must process.

The second line of input contains n space-separated integers $a_1, a_2, a_3, \ldots, a_n$, describing which entries from the Fibonacci sequence are written on each amulet.

Then, q lines follow, each describing a query. Each line contains the three space-separated integers L, R, and x for that query.

Output Format

Output q lines, answering each of the queries in order.

Constraints and Subtasks

For all subtasks
$1 \leq n, q \leq 123456$ $0 \leq a_i \leq 10^{18}$ In all queries, $R - L + 1 \geq 4$.

Subtask	Points	Constraints
1	**10**	$n, q \leq 70$ $a_i \leq 40$
2	**25**	$n, q \leq 70$
3	**25**	$n, q \leq 10^5$ $a_i \leq 40$
4	**20**	$n, q \leq 10^5$ In all queries, $L = 1$ and $R = n$
5	**15**	$n, q \leq 10^5$
6	**5**	No further constraints.

Sample I/O

Input 1	Output 1

`5 2`
`1 3 5 2 7`
`1 5 20`
`1 5 2`
	`5`
	`0`

Input 2	Output 2

`8 2`
`5 6 7 8 8 7 6 5`
`1 8 10`
`2 7 10`
	`60`
	`9`

Explanation

For the first sample input, with $a = \{1, 3, 5, 2, 7\}$, the actual values of the amulets are $\{1, 2, 5, 1, 13\}$.

For the queries:

1. We have $L = 1$ and $R = 5$, and Li's amulet has value $F_{20} = 6765$
2. We have $L = 1$ and $R = 5$, and Li's amulet has value $F_2 = 1$

The Manga Guide to Competitive Programming

In an attempt at reaching out to the interests of the youth, NOI.PH has commissioned fan-artist and friend Saji to create, "The Manga Guide to Competitive Programming." We plan to sell it at a booth at the next Komiket, hoping to convert more people to the joys of competitive programming!

Saji is setting up her Komiket booth. Her booth will display a row of n stacks of books, indexed from 1 to n. Initially, each stack has 0 books.

We cannot put too many books in a particular stack, otherwise we might obscure our seller, or our posters, or the booths behind us, or it could just be a safety hazard. The maximum capacities of each stack are described by $a_1, a_2, a_3, \ldots, a_n$, meaning that stack i should contain **strictly less than** a_i books at all times. Thus, the number of books in stack i can be any integer from 0 to $a_i - 1$.

We only sell one book, so the *state* of our booth can be described by the number of books in each stack. Saji wondered how many possible different states her booth could be in, at any given time. Wanting to impress her, you explain that it's just simple combinatorics: For each i, there are a_i possible values for the number of books in stack i, and these are all independent of each other, and therefore there should be a total of $a_1 \times a_2 \times a_3 \times \cdots \times a_n$ different states overall.

Saji *understands* your argument at an intellectual level, but admits that she can't visualize it or intuitively "get" it. Okay, looks like we'll need to show her in order to truly convince her. Seeing is believing!

Most programmers are wimpy nerds, so let's assume that in one move, you can only add or remove one book from one stack. In other words, you only have the following two operations available to you.

- "`ADD i`" to **add** a single book to stack i.
- "`RM i`" to **remove** a single book from stack i.

The booth begins with each stack having 0 books. Write a program that, given the values of $a_1, a_2, a_3, \ldots, a_n$, outputs some sequence of the operations described above, that accomplishes the following three goals:

- As the operations are performed, the booth must visit each of the $a_1 \times a_2 \times a_3 \times \cdots \times a_n$ possible states that it could be in.
- We need to pack up at the end of the day. When your program terminates, each of the stacks must again contain 0 books.
- To be efficient, you must perform the *minimum* number of operations to achieve both of the previous two goals.

Input Format

The first line of input contains a single integer n, the number of stacks.

The second line contains n space-separated integers $a_1, a_2, a_3, \ldots, a_n$, describing the maximum capacities of each stack.

Output Format

First, output a line containing a positive integer k, the number of operations your program is going to perform.

Then, output k lines, each of the form `ADD i` or `RM i`, where $1 \leq i \leq n$, corresponding to the operations to add or remove a book from stack i, respectively. If you call `ADD i` on a stack and this causes it to have a_i books, or if you call `RM i` on a stack with 0 books, then you will immediately receive a Wrong Answer verdict.

If the first two goals are accomplished after the k operations are executed *and* k is minimized (accomplishing the third goal), then you will receive an Accepted verdict. If there are multiple solutions, any that accomplish all three goals will be accepted.

Constraints and Subtasks

For all subtasks
$1 \leq n$ $2 \leq a_i$ for all i $a_1 \times a_2 \times a_3 \times \cdots \times a_n \leq 10^5$

Subtask	Points	Constraints
1	**22**	$n \leq 2$
2	**13**	$n \leq 3$
3	**12**	$a_i = 2$ for all i
4	**11**	$n = 6$ and a is some permutation of $\{2, 3, 5, 7, 11, 13\}$
5	**17**	$a_i = 3$ for all i
6	**25**	No further constraints.

Sample I/O

Input	Output

`2`
`3 4`
	`12`
	`ADD 2`
	`ADD 1`
	`ADD 2`
	`RM 1`
	`ADD 2`
	`ADD 1`
	`ADD 1`
	`RM 2`
	`RM 2`
	`RM 2`
	`RM 1`
	`RM 1`

Explanation

The following diagram shows the states visited by the sample output as it goes through the described operations. It visits all $3 \times 4 = 12$ states and then finishes on the state with 0 books in any stack. We can show that $k = 12$ is the minimum number of operations needed to accomplish the given task.