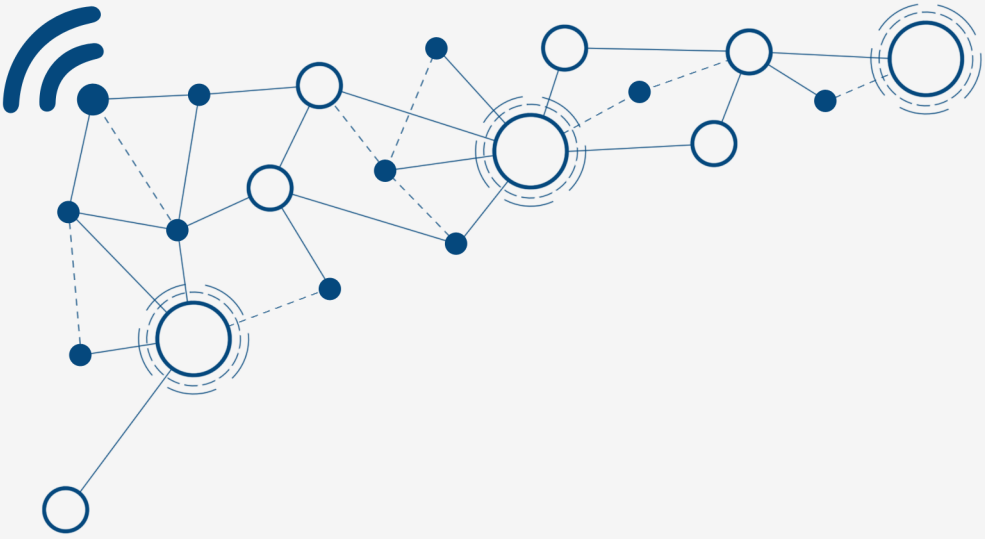
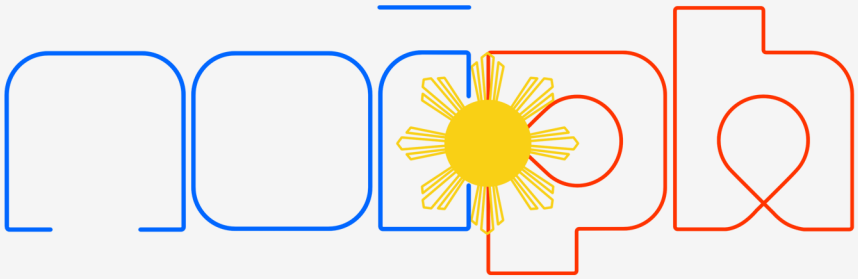
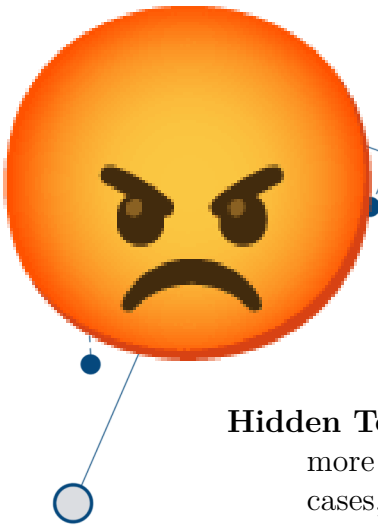




2026

National Olympiad in Informatics

Elimination Round - WAVE 3



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` with `flush=True`
- For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the contest! 😃

Contents

Problem A
Penrick Gets Disqualified

Penchick has become NOI.PH's mascot, a friend to all competitive programmers.

There are many penchicks in this world, each with their own name. But some have an evil alter ego, always known as **Penrick**.

`:penrick:`

One penchick unfortunately was caught breaking the official NOI.PH Elims rules.

Penchick tried to argue that it was not them who cheated, but their Penrick side. How could they be punished for Penrick's transgressions!? Sadly, possession by an evil alter ago is not a valid excuse for breaking the rules (there *is* no excuse).

You will be given two names in the input—one of them is `Penrick`, and the other one is the true name of this penchick (though not necessarily in that order). Please output the name of who will face the consequences of Penrick's cheating.
(Hint: It isn't Penrick)

Also, Penrick may try to confuse you. All names *should* begin with a capital letter, with all other letters being lowercase. Penrick may have messed with the cases of the letters in the input to try to disguise themselves; how dastardly!

Please fix it before outputting your answer.

Input Format

Input consists of a single line containing two space-separated names—one of these is Penrick's, and the other one is the penchick's true name (in some order). The cases may or may not have been tampered with.

Output Format

Output the name of who will face the consequences for Penrick's misdeeds.

Constraints and Subtasks

Constraints are promises that are guaranteed to be satisfied by all test input that your program will be tested on. Your submission does not need to validate that the input has these properties (we promise that they always already do).

For all subtasks
Exactly one of the names in the input is `Penrick` (once the cases are fixed) Each name consists of at least 1 and at most 10 upper or lowercase English letters.

Subtasks are additional special constraints that certain test cases may also follow. If you correctly answer all test cases belonging to a certain subtask, then you are awarded that subtask's points (allowing you to still earn points even if you cannot answer the full version of the problem).

Subtask	Points	Constraints
1	**25**	The names' cases are correct. Penrick's name is given first.
2	**25**	The names' cases are correct.
3	**25**	Penrick's name is given first.
4	**25**	No further constraints. *(Solve the full version of the problem to get these points)*

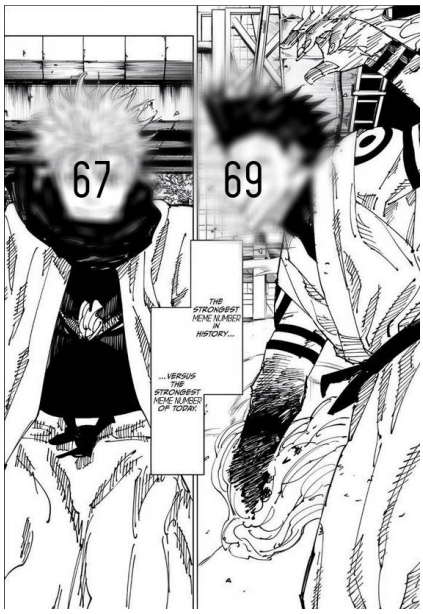
Sample I/O

Input	Output
`Penrick Penchick`	`Penchick`

Input	Output
`PeLiCaN penRICK`	`Pelican`

Problem B

The Strongest Meme Number in History vs The Strongest Meme Number of Today



The **subset sum** problem is a classic *decision problem* in computer science.

Given a list of n integers $a_1, a_2, \ldots, a_n$ and some integer x, the problem asks whether or not there exists a (possibly empty) *subset* of the items in your list such that the sum of the numbers in this subset is exactly x.

For example, if $a = [67, 67, 67, 69]$, then the answer would be `Yes` for $x = 136$ (taking $[67, 69]$) and $x = 0$ (taking $[]$), but `No` for $x = 100$.

Subset sum is NP-complete, so solving it in polynomial time means you can also answer *all* other problems within the NP complexity class in polynomial time.

No known algorithm can solve the general problem in polynomial time, and if one were to be found, then it would be equivalent to proving P=NP, one of the seven Millennium Prize problems.

But wait, there's more! We can consider a *counting* version of the problem!

Given a list of n integers $a_1, a_2, \ldots, a_n$, your **true task** for this problem is to **count** the number of integers x such that the answer to the subset sum problem (on this a, with this x) would be `Yes`! And you must do it in polynomial time!

Counting problems usually belong to the ♯P complexity class, and oftentimes no known polynomial time algorithm is known for such problems, even when one is known for its decision version (for example: counting vs finding toposorts).

But I believe in you, programmer! You can do it!

Hint: Pay *veeeeery* close attention to the constraints.

Input Format

The first line of input contains a single integer n.

The second line of input contains the n space-separated integers $a_1, a_2, \ldots, a_n$.

Output Format

Output the number of integers x for which: there exists a subset of the items in n which sum up to that x.

Constraints and Subtasks

Constraints are promises that are guaranteed to be satisfied by all test input that your program will be tested on. Your submission does not need to validate that the input has these properties (we promise that they always already do).

For all subtasks
$1 \leq n \leq 2 \times 10^5$ $a_i = 67$ or 69 for all i.

Subtasks are additional special constraints that certain test cases may also follow. If you correctly answer all test cases belonging to a certain subtask, then you are awarded that subtask's points (allowing you to still earn points even if you cannot answer the full version of the problem).

Subtask	Points	Constraints
1	**2**	$a_i = 67$ for all i.
2	**33**	$n \leq 15$
3	**34**	$n \leq 2026$
4	**31**	No further constraints. *(Solve the full version of the problem to get these points)*

Sample I/O

Input	Output
4 67 67 67 69	8

Explanation

In the sample input, the valid x are $\{0, 67, 69, 134, 136, 201, 203, 270\}$.

Problem C

Leo/need's Friendship Bracelets

Under the light of the Leonid Meteor Shower, best friends and bandmates Melody Hoshino, Daffodil Tenma, Bituin Mochizuki, and Sunny Hinomori were making *friendship bracelets* to reaffirm their friendship after a certain series of events.

Each friendship bracelet consists of a positive integer number of beads — after all, having no beads on a bracelet is "no fun!!! 🥺", according to Daffodil Tenma. Furthermore, Melody Hoshino wanted a common theme to each friendship bracelet — after all, if they were to remain good friends with one another, they must also ensure that their bracelets complemented one another.

To incorporate this suggestion, Bituin Mochizuki suggested that they try to make *familiar pairs* of friendship bracelets.

Bituin Mochizuki clarifies that two friendship bracelets with p and q beads form a *familiar pair* if and only if: p divides q; or q divides p. She reasons that, if this condition holds, then one bracelet fits into another, pointing toward an intimate connection between bracelets that mirrors that of their owners'.

Sunny Hinomori thus decided to keep track of the number of familiar pairs *over time*, as new friendship bracelets are being made.

Formally, suppose that the friends make n bracelets (where n is a positive integer), and that the ith created bracelet has a_i beads. Then, for every t from 1 to n, Sunny Hinomori computes the number of pairs (i, j) satisfying $1 \leq i < j \leq t$ such that bracelets i and j (with a_i and a_j beads respectively) form a familiar pair.

However, Sunny Hinomori started feeling tired doing this — she wanted to focus on spending time with her friends, not on counting the number of familiar pairs! Could you write a program to help her compute these values, so that Sunny could spend some quality time with her friends?

Input Format

The first line of input contains a single integer n.

The second line of input contains the n space-separated integers $a_1, a_2, \ldots, a_n$.

Output Format

Output a line of n space-separated integers, where the t'th of these is the number of familiar pairs among the first t friendship bracelets only.

Constraints and Subtasks

For all subtasks
$1 \leq n \leq 3 \times 10^5$ $1 \leq a_i \leq 3 \times 10^5$ for all i

Subtask	Points	Constraints
1	**14**	$n \leq 500$
2	**22**	$n \leq 2000$
3	**20**	$a_i = i$ for all i
4	**21**	$n \leq 10^5$ $a_i \leq 1000$ for all i
5	**17**	$n \leq 10^5$ $a_i \leq 10^5$ for all i
6	**6**	No further constraints.

Sample I/O

Input	Output
6 3 1 4 1 5 9	0 1 2 5 7 10

Input	Output
6 1 2 3 4 5 6	0 1 2 4 5 8

Input	Output
4 1 10 100 200	0 1 3 6

Problem D
Rectangular Painting at 25:00

Enanan is creating cover art for *Discord at 25:00*'s upcoming new song, Theatre, which was composed by her friend K in dedication to her friend Yuki. Enanan, being good friends with Yuki, wished to draw something that authentically expresses Yuki's feelings.

Amia, a friend of Yuki, suggested that Enanan use $p \times q$ rectangles in her art piece. Enanan immediately saw the vision—out of the perfect right-angled facade of the rectangle, she could depict an emergent imperfection that mirrors Yuki's wish to forego her mask of perfection and express her true self.

Enanan drafted an artwork on *AlabAlpaka* and proceeded to find a construction of rectangles that mimics her art. However, this proved to be a difficult problem, and so she enlisted the help of Sir K (not to be confused with her friend K)!

Thankfully, Sir K was able to reframe Enanan's conundrum as follows:

First, since Enanan's painting is monochrome, each color could be represented as a (possibly negative) integer. Initially, Enanan's painting is composed of nm cells, arranged in a grid of n rows and m columns.

Enanan's AlabAlpaka artwork also consists of nm cells arranged in a grid of n rows and m columns. The cell in the ith row from the top and jth column from the left of her AlabAlpaka artwork is denoted by (i, j), and its color is the integer $R_{i,j}$, **initially all** 0 in the editing software.

Enanan wants her AlabAlpaka artwork to match her painting. Specifically, she wants $R_{i,j}$ to have the value $A_{i,j}$ for all i, j. As Enanan may only paint in $p \times q$ rectangles (which **cannot** be rotated!), she may perform the following operation on her painting at most 10^6 times:

Paint a Subrectangle. Enanan may pick any $p \times q$ subrectangle of her painting and add the *same* value k of her choice to each cell within it. Specifically...

- She may choose two positive integers i, j and an integer k satisfying:
  - $1 \le i \le n - p + 1$
  - $1 \le j \le m - q + 1$
  - $|k| \le 10^{12}$
- Then, she adds this same value k to $R_{i',j'}$ for every cell (i', j') where:
  - $i' \in [i, i + p - 1]$
  - $j' \in [j, j + q - 1]$

Enanan thanked Sir K for all his help; however, Enanan *did not sign up for all of this math*! Thus, she has asked *you* for help.

Given her AlabAlpaka artwork, could you construct a sequence of **Paint a Subrectangle** moves that she could do to replicate her AlabAlpaka artwork? If the task is impossible, please also tell Enanan so that she could quickly come up with a backup plan!

Input Format

The first line of input contains four space-separated positive integers n, m, p, and q, the number of rows and columns of A, and the number of rows and columns of the subrectangle, respectively.

Then, n lines follow, each with m space-separated integers, encoding the desired colors in the artwork. The jth integer in the ith line corresponds to $A_{i,j}$.

Output Format

Output a single line, containing only the word `YES` or `NO`. If the AlabAlpaka artwork A can be constructed using the operations described, your output must be `YES`. Otherwise, it must be `NO`.

Furthermore, if your answer is `YES`, you must describe the sequence of moves that constructs A. First, output a single nonnegative integer ℓ in the next line — the number of moves required. Note that $0 \le \ell \le 10^6$ must hold.

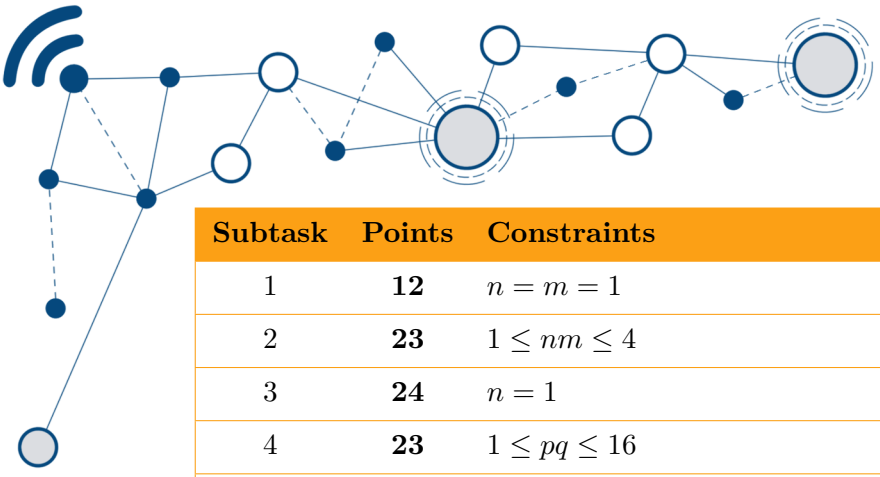
Then, output ℓ lines, each containing three space-separated integers i_ℓ, j_ℓ, k_ℓ, in that order. This represents adding k_ℓ to each element in the $p \times q$ subrectangle whose top-left cell is at (i_ℓ, j_ℓ).

Note that $1 \le i_\ell \le n - p + 1$, and $1 \le j_\ell \le m - q + 1$, and $|k_\ell| \le 10^{12}$ must hold for each ℓ.

It can be shown that if a solution exists, then there exists one satisfying the aforementioned constraints. Also, you **don't** need to minimize ℓ.

Constraints and Subtasks

For all subtasks
$1 \le n, m$ $1 \le nm \le 2 \times 10^5$ $1 \le p \le n$ $1 \le q \le m$ $\lvert A_{i,j} \rvert \le 10^6$ for each (i, j)



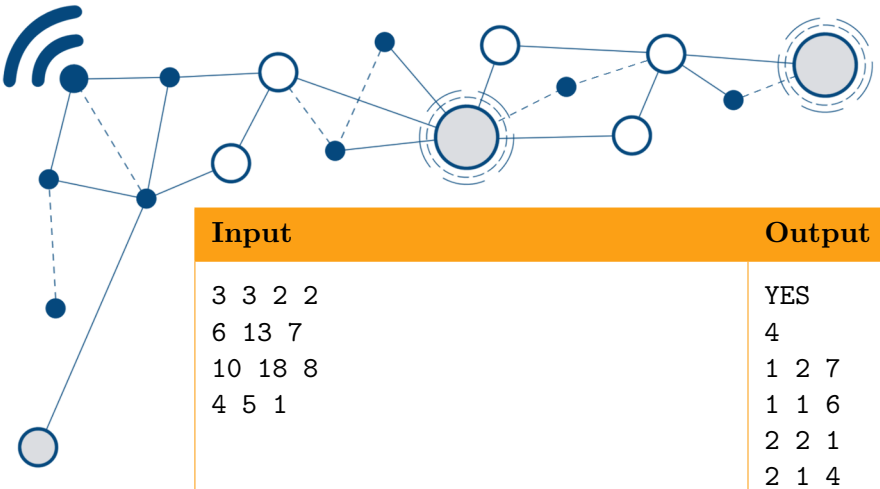
Subtask	Points	Constraints
1	**12**	$n = m = 1$
2	**23**	$1 \leq nm \leq 4$
3	**24**	$n = 1$
4	**23**	$1 \leq pq \leq 16$
5	**18**	No further constraints.

Sample I/O

Input	Output
4 3 2 3 3 3 3 7 7 7 9 9 9 5 5 5	YES 3 2 1 4 3 1 5 1 1 3

Input	Output
4 3 2 3 3 3 3 7 7 7 9 9 9 4 4 4	NO

Input	Output
1 1 1 1 1000	YES 1 1 1 1000



Input	Output
3 3 2 2 6 13 7 10 18 8 4 5 1	YES 4 1 2 7 1 1 6 2 2 1 2 1 4

Input	Output
3 3 1 2 0 0 0 0 0 0 0 0 0	YES 0

Problem E
The Woke Mind Virus

We have a moral crisis on our hands. All of our children are becoming *woke*!

By that, of course, I mean that they're suffering from insomnia. Actually, most of these teenagers don't really have insomnia, just really irresponsible sleep schedules. But this is a crisis nonetheless!

We have decided to call it the *Woke Mind Virus*, and it's infecting more and more people each day. You never know who could be next... It could be your friend, or it could even be you!

A certain HS class has the seats in its classroom arranged into a grid of 2 rows and n columns. Initially, only the top-left-most student has the Woke Mind Virus.

Every day, an Agent of Evil selects one of the $2n$ students uniformly at random:

- Suppose the selected student does not have the Woke Mind Virus...
  - If they are next to someone with the Woke Mind Virus ("sharing a corner" is not enough; they must be *directly* north/south/east/west of an infected person), then they get the Woke Mind Virus too.
  - Otherwise, they fight off temptation and do **not** get infected.
- If the selected student already has the Woke Mind Virus, then nothing happens (they are *already* infected after all).

On average, how many days do we **expect** to need to wait until all students have the Woke Mind Virus? And just to stop any dastardly strategies, we're going to ask for it modulo p, where p is a prime given in the input as well.

Feel free to look up how to handle rational numbers modulo a prime. It's a recurring concept in many competitive programming probability problems!

Suppose we repeatedly perform the random experiment of "repeatedly select random students until everyone is infected" and then record how many days it took. The expected value is equal to what our running average will converge to (with high probability).

*Formally, let $x_1, x_2, x_3, \ldots$ be an infinite sequence where x_t is the **average** of the recorded number of days from the first t random experiments. We can prove that there exist positive integers a and b such that $\lim_{t\to\infty} x_t = a/b$ almost surely.*

Your task is to report the value of $ab^{-1} \mod p$; more formally, output the value r such that $a \equiv br \mod p$. We can show that such an r always exists and is unique (modulo p), given the constraints.

Input Format

Input consists of a single line containing the two space-separated integers n and p.

Output Format

Output the expected value modulo p.

Constraints and Subtasks

For all subtasks
$1 \le n \le 10^4$ $2n < p \le 2 \times 10^9$.

Subtask	Points	Constraints
1	**39**	$n \le 11$
2	**28**	$n \le 200$
3	**8**	$p = 2 \times 10^6 + 3$
4	**25**	No further constraints. *(Solve the full version of the problem to get these points)*

Sample I/O

Input	Output
1 19	2

Input	Output
2 19	8

Explanation

In the first sample input, the expected value is 2 days on average.

There is one infected person, and one other student who isn't.

- $1/2$ chance that the other student is first infected on Day 1.
- $(1/2)^2$ chance that the other student is first infected on Day 2.
- $(1/2)^3$ chance that the other student is first infected on Day 3.
- ...and so on.

Because $1/2 + 2/2^2 + 3/2^3 + \cdots = 2$, the expected value is 2 exactly. Since it is already an integer, 2 mod 19 is still 2.

Problem F
The Pisano Dynasty

Many years ago, the Pisano sisters, aka the Fibonacci Five-tuplets, competed against one another to win the heart of Simoy, their shared tutor. Eventually, [redacted] triumphed, and she and Simoy married and lived happily ever after.

Now, with [redacted] as the matriarch, the Pisano clan has grown to have n people in total. Their family reunions have some very fun Fibonacci-themed party games! Shall we observe one?

You are reminded that the Fibonacci numbers $F_1, F_2, F_3, F_4, \ldots$ are an infinite sequence defined recursively as follows.

- $F_1 = F_2 = 1$
- For $n > 2$, we let $F_n = F_{n-1} + F_{n-2}$.

The sequence begins $1, 1, 2, 3, 5, 8, 13, 21, \ldots$

The n people in the Pisano clan line up in a row, and index themselves from 1 to n. Everyone holds a Pisano paper which contains a Fibonacci number. Let p_i denote the number currently written on the Pisano paper of family member i; initially, **all** Pisano Papers display the number 1, i.e. $p_i = 1$ for all i.

Then, q times, the game master [redacted] performs one of two kinds of operations:

- "`? l r`" means that **only** Pisano clan members ℓ through r (inclusive) participate; everyone else steps aside for now. They do this:
  - All distinct (unordered) pairs of participating Pisano clan members shake hands, and when they do, they write down the product of the Fibonacci numbers currently written on their Pisano papers.
  - Then, our game master [redacted] computes the sum of all $\binom{r-\ell+1}{2}$ numbers that were written down, taken modulo 998244353. Output it.
- "`! l r`" means that we will **update** the Pisano papers of clan members ℓ through r with new Fibonacci numbers, via the following process:
  - $p_\ell = F_1$
  - $p_{\ell+1} = F_2$
  - $p_{\ell+2} = F_3$
  - $p_{\ell+3} = F_4$
  - $\vdots$
  - $p_r = F_{r-\ell+1}$

Please simulate such an instance of this party game for us! It sounds like fun!

Input Format

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

Then, q lines follow, each describing an operation. Each operation is in one of the two formats described above.

Output Format

For each `?` operation, output a line containing the value that the game master computed during that operation.

Constraints and Subtasks

For all subtasks
$1 \leq n \leq 10^{18}$ $1 \leq q \leq 10^5$ $1 \leq \ell \leq r \leq n$ in each operation

Subtask	Points	Constraints
1	**11**	$n \leq 20$ $q \leq 100$
2	**11**	$n \leq 500$ $q \leq 500$
3	**12**	$n \leq 3000$ $q \leq 3000$
4	**20**	$r - \ell \leq 10^7$ There are at most 100 `!` operations.
5	**7**	$r - \ell \leq 10^7$ $q \leq 1000$
6	**10**	$r - \ell \leq 10^7$
7	**29**	No further constraints.

Sample I/O

Input	Output
10 5 ? 1 10 ! 3 8 ? 6 10 ! 1 6 ? 1 7	45 112 248

Explanation

For the second operation, we update members 3 through 8, so the Pisano papers contain these Fibonnaci numbers (where the updated values are colored red):

$$1, 1, \color{red}{1, 1, 2, 3, 5, 8,}1, 1,$$

For the third operation, we take only the clan members 6 through 10, so the relevant Pisano papers have the values $3, 5, 8, 1, 1$.

There are going to be ten handshakes, and they will write these values:

- $3 \times 5 = 15$
- $3 \times 8 = 24$
- $3 \times 1 = 3$
- $3 \times 1 = 3$
- $5 \times 8 = 40$
- $5 \times 1 = 5$
- $5 \times 1 = 5$
- $8 \times 1 = 8$
- $8 \times 1 = 8$
- $1 \times 1 = 1$

In total, $15 + 24 + 3 + 3 + 40 + 5 + 5 + 8 + 8 + 1 = 112$, so that is the answer to the third operation.

Problem G
Spam and Hans

Remember Hans from 2023? He was caught cheating and thus was disqualified from the 2023 NOI.PH Eliminations. Wishing to avenge himself, Hans continued with his treacherous acts. Over four long years, he has been studying Large Language Models (LLMs) in the hopes of toppling NOI.PH.

Unfortunately, since he relied too much on GPT to learn about LLMs (yes, he used an LLM to learn about LLMs), he did not understand anything! It seems his plan would not come to fruition...

A few years later, when a penchick named Chubby became the mascot of the first Penchick Online Math Olympiad (POMO1), his brother Pelican became jealous. His jealousy turned him into Penrick, a narcissistic, attention-seeking, villainous alter-ego of himself (this sounds familiar).

Seeing how penchicks took over NOI.PH by storm, he was filled with animosity. Together with Hans, he devised a truly terrible invention to destroy NOI.PH: The Penchick Spammer! The Penchick Spammer is an abominable amalgamation of Hans' LLMs and the Penchick Replication technique. They plan to spam evil text into NOI.PH problems!

Specifically, the Penchick Spammer maintains a **zero-indexed** string T, **initially empty**. Then, in one step, Hans and Penrick may insert k copies of the string S at index i of T. We clarify that, after this move, the index of the first character of the first instance of S in T will be i.

An example should help clarify. Recall that T starts empty. Then, you could do these four insert operations on it, with the following respective (i, k, S) values:

i	k	S	Resulting T
0	2	`su`	`susu`
4	1	`s`	`susus`
3	1	`susamog`	`sussusamogus`
1	3	`spam`	`sspamspamspamussusamogus`

Table 1: Example moves

Hans' noble twin brother OganessonDecay628 learned about this plot and informed NOI.PH about it. NOI.PH decided to take matters into their own hands to ensure that Penrick and Hans would not corrupt their problem statements. However, they need your help to counteract Penrick and Hans' attacks!

Could you determine, at any point in time, what the ith character in the string T maintained by the Penchick Spammer is? Quick, there is not much time, Penrick and Hans seem to be activating the machi- AAAAAAAAHACKEDBYPENRICKANDHANSHACKEDBYPENRICKANDHANS

Input Format

The first line of input contains an integer q.

Then, q lines of input follow. Each line is of the form:

1. "`I i k S`" indicating that Penrick and Hans insert k instances of the string S into index i.
2. "`Q i`" indicating you must output the letter in string T at index i (you are reminded that T is zero-indexed) at this point in time.

These q events must be processed in order.

Output Format

Output a line for each `Q` type query, where each contains the answer to that query.

Constraints and Subtasks

For all subtasks
$1 \leq q \leq 2 \times 10^5$ $1 \leq k \leq 10^8$ The sum of $\lvert S \rvert$ across all queries is $\leq 2 \cdot 10^5$ Each S consists of lowercase English letters. For each query type `I`, $0 \leq i \leq \lvert T \rvert$ at that point in time For each query type `Q`, $0 \leq i \leq \lvert T \rvert - 1$ at that point in time

Subtask	Points	Constraints
1	**21**	$q \leq 1000$ $i \leq 1000$ in all queries.
2	**27**	$q \leq 1000$
3	**23**	$k = 1$ in all queries.
4	**29**	No further constraints.

Sample I/O

Input	Output
24 I 0 1 na I 2 2 bag I 2 2 ka I 6 2 pa I 8 1 g I 14 1 a Q 0 Q 1 Q 2 Q 3 Q 4 Q 5 Q 6 Q 7 Q 8 Q 9 Q 10 Q 11 Q 12 Q 13 Q 14 Q 15 Q 16 Q 17	n a k a k a p a g p a b a g a b a g

Problem H
Drinking History

On today's episode of *Drinking History*, we'll be revisiting some vintage bottles of wine from my personal collection. These kinds of wine were consumed by the Spanish during their historical occupation of the Philippines. In fact, some of these bottles are up to hundreds of years old!

I have n bottles of vintage wine. You are assured that every wine in my collection is from a **distinct** year. It would be nice if, even in just the thumbnail of the YouTube video, I could already start to showcase the bottles in my collection in an aesthetically pleasing way.

I would like to choose some subset of my bottles and arrange them *in a row*, such that for every pair of adjacent bottles: the last two digits on the year of the previous bottle *exactly equals* the first two digits on the year of the next bottle.

So for example, this would be valid: `1916`, then `1620`, then `2015`.

What is the maximum number of bottles I can put on the thumbnail? Help me construct the thumbnail too, please—which bottles should I use, and in what order?

Oh I'm sorry, do you think this problem statement is lacking in flavor? Dear viewer, for today's problem, the flavor is distributed evenly throughout the problem statement, not just in the story!

Input Format

The first line of input contains a single integer n.

The second line of input contains the n space-separated integers, the years on the bottles.

Output Format

First, output a positive integer k, the maximum number of bottles that can be displayed.

Then, output k space-separated integers. Each should be a year that belongs to a bottle that I own (where no bottle may be used more than once), and the last two digits on the year of each bottle (except the last) should match the first two digits on the year of the bottle that comes after it.

If there are multiple possible solutions, any will be accepted (as long as k is maximized).

Constraints and Subtasks

For all subtasks
The bottles' years are all distinct integers. The most recent year in my collection is not later than the year Pope Francis visited the Philippines (in 2015).

Subtask	Points	Constraints
1	**2**	The oldest year in my collection is not earlier than the year of the EDSA People Power Revolution (in 1986).
2	**29**	The oldest year in my collection is not earlier than the year the Philippines declared independence from Spain (in 1898).
3	**22**	The oldest year in my collection is not earlier than the year that the revolt of Gabriela and Diego Silang began (in 1762).
4	**8**	The oldest year in my collection is not earlier than the year of St Lorenzo Ruiz's death and martyrdom (in 1637).
5	**39**	The oldest year in my collection is not earlier than the year that Magellan first set foot on the Philippines (in 1521).

Sample I/O

Input	Output
4 1620 1916 2008 2015	3 1916 1620 2015

Input	Output
5 1521 1621 1721 1821 1921	1 1521

Explanation

In the first sample input, an answer of "1916 1620 2008" would also have been accepted.

In the second sample input, an answer of "1621" would also have been accepted (among others).

z

Problem I
The Apotheosis Is Upon Us

Paul wakes up in his home town of Hatchetfield to find that everyone around him has started... singing. And dancing. Like they're in a musical! And for a guy who absolutely does *not* like musicals, Paul needs to get away from it all.

He goes to his favorite coffee shop, Beanie's, and in a bid to feel like all this is a bad dream, he orders a cup of black coffee to wake him up. To his horror, all but one of the baristas start serving everyone poisoned coffee, and as they take a sip, the customers in Beanie's start singing as well. You might say that this doesn't sound that scary, but it is if you think about the *implications*.

Paul escapes from Beanie's with Emma, the only other non-infected person in the shop. In the alley behind Beanie's, the two of them run into some of Paul's office workers, bringing the group to k people. They need to get to Emma's kooky, reclusive biology professor who can analyze the poison and and understand what's going on, but as the sun sets, they need to try to come up with a plan to hide from all the singing and dancing.

Hatchetfield can be represented as a set of m roads, each connecting two of n buildings (which are labeled 1 to n). In order to keep each other safe, they need to select k buildings and $k-1$ roads, such that the k buildings they selected are reachable from one another using only the $k-1$ roads they selected. This way, they can still reach each other even if every other road starts getting infected by the music.

How many ways are there for them to come up with such a plan? Two plans are distinct if there is a road or building included in one plan but not in the other. Quickly, lest their *apotheosis* be upon them!

Input Format

The first line of input contains two space-separated integers n, m, and k, each representing the number of buildings, the number of roads, and the number of people in Paul's group.

Then, m lines follow. The ith line contains two space-separated integers u_i and v_i, representing the two buildings connected by road i. Here, the buildings are integers from 1 through n.

Output Format

Output the number of plans satisfying the conditions specified in the problem statement.

Constraints and Subtasks

For all subtasks
$2 \le k \le n \le 2 \cdot 10^5$ $0 \le m \le \binom{n}{2}$ $u_i \ne v_i$ for all $1 \le i \le m$ $\{u_i, v_i\} \ne \{u_j, v_j\}$ for all $1 \le i < j \le m$

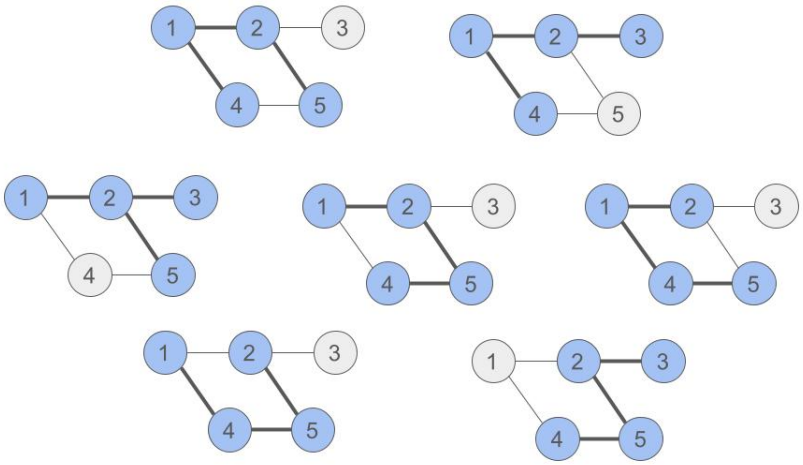
Subtask	Points	Constraints
1	**2**	$k = 2$ $m \le 2 \cdot 10^5$
2	**5**	$k \le 3$ $m \le 2000$
3	**10**	$k \le 3$ $m \le 2 \cdot 10^5$
4	**6**	$k \le 4$ $m \le 20$
5	**19**	$k \le 4$ $m \le 2000$
6	**11**	$k \le 4$ $m \le 2 \cdot 10^5$
7	**7**	$k \le 5$ $m \le 20$
8	**26**	$k \le 5$ $m \le 2000$
9	**14**	$k \le 5$ $n \le 4000$ $m \le 2 \cdot 10^5$

Sample I/O

Input	Output
5 5 4 1 2 2 4 5 4 5 1 3 5	7

Explanation

Here are all the possible ways to select three roads and four buildings:



Problem J
Make a New Theme-CP

Stack Tracen Academy is the premier national institution for training students in the art of competitive progra-uma-ing[1]. Stack Tracen enjoys hosting exciting new events with various gimmicks, in order to inspire their students to twinkle and become legends!

Here is a totally irrelevant list of gimmicks that the Japanese branch of Stack Tracen Academy has used for training in the past:

- Comp prog training, but you have to do sports while coding
- Comp prog training, but you have to cook nourishing dishes while coding
- Comp prog training, but in France
- Comp prog training, but in America
- Comp prog training, but in an onsen
- Comp prog training, but in a deserted island

The international Stack Tracen's latest gimmick is simple—there are no pre-defined paths. Students should blaze their own track and decide how they want to train!

The training program in this event will last n days, where on each day, you can choose whether or not you want to do the provided Theme CP[2] training contest on that day. If you choose to do the Theme CP on the ith day, you earn a_i points. If you don't do the Theme CP on the ith day, you **may not** come back to it later. The goal is to earn as many points as you can.

That's amazing! Why not just do a Theme CP every single day, then!? Well, we don't want you to suffer Comp Prog Fatigue, so we impose the following limitation:

- You may not do a Theme CP for three days in a row.

Given this restriction, what is the maximum number of points that you can obtain if you train optimally?

What's the catch? I hope you didn't think this training program was going to be easy. You see, n can be very, *very* large...

Input Format

We encode $a_1, a_2, a_3, \ldots, a_n$ in an unusual way, using another sequence b of length m.

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

The second line of input contains the m space-separated integers $b_1, b_2, b_3, \ldots, b_m$.

[1] I'm trying my best here, okay

[2] A system developed by an NOI.PH alum!

Now, let a be formed by periodically repeating b until we get n elements. Formally, for each $1 \le i \le n$, let $a_i = b_{((i-1) \bmod m)+1}$

Output Format

Output a single integer, the maximum number of points that could be attained over the n days.

Constraints and Subtasks

For all subtasks
$1 \le n \le 10^{15}$ $1 \le m \le 2 \times 10^5$ $1 \le b_i \le 1000$ for each i.

Subtask	Points	Constraints
1	**25**	$m = 1$
2	**20**	$n \le 2000$
3	**20**	$m \le 2000$
4	**7**	$n \le 2 \times 10^5$
5	**28**	No further constraints.

Sample I/O

Input	Output
10 4 5 2 3 4	26

Explanation

In the sample input, $a = [5, 2, 3, 4, 5, 2, 3, 4, 5, 2]$.

We can train on the 1st, 2nd, 4th, 5th, 7th, 9th, and 10th days, i.e. claim the points highlighted red in the following schedule:

$$[5, 2, 3, 4, 5, 2, 3, 4, 5, 2]$$

The total points of $5 + 2 + 4 + 5 + 3 + 5 + 2 = 26$ is optimal.