icpc International Collegiate Programming Contest

The 2024 ICPC Pacific Northwest Regional Contest

Problem Set

icpc.foundation

icpc Pacific Northwest

JETBRAINS

icpc global sponsor
programming tools

upsilon pi epsilon
honor society

Pacific Northwest Regional Programming Contest
Division 1

16 November 2024

- The languages supported are C, C++ 20 (with Gnu extensions), Java, Python 3 (with pypy3), and Kotlin.
- Python 2 and C# are not supported this year.
- For all problems, read the input data from standard input and write the results to standard output.
- In general, when there is more than one integer or word on an input line, they will be separated from each other by exactly one space. No input lines will have leading or trailing spaces, and tabs will never appear in any input.
- Submit only a single source file for each problem.

A: Auto-Coin-o-Matic

Time Limit: 2 seconds, Memory limit: 2G

It's finally here! The day you unveil your new invention, the Auto-Coin-o-Matic! You watch with glee and anxiety as people insert their card into the machine, type in the amount they want, and get exact change out with the fewest number of coins.

But was it actually the fewest number of coins? That's how it was programmed, but what if you had a bug? It's okay, you're watching. You decide to randomly pick some transactions and double check that what the machine gave out is indeed the fewest number of coins possible. But, oh no, the machine is running out of certain types of coins! Will it still work correctly?

Input

The input starts with two integers n and m ($1 \leq n \leq 2000$, $1 \leq m \leq 10^5$).

The next line contains n integers, $d_1, d_2, \ldots, d_n$ ($1 \leq d_i \leq 10^5$) representing the denominations of coins available initially. It is guaranteed that all denominations are unique.

The next m lines contain a character c ($c \in \{\texttt{Q}, \texttt{X}\}$) and an integer v ($1 \leq v \leq 10^5$), where c is the type of event and v is the value of the event.

- If c is the character `Q`, this is a query and the output should be the minimum number of coins needed to give out exactly v. It is guaranteed that there will be at least one query.

  If it is not possible to make v exactly with the available denominations, output -1 instead.

- If c is the character `X`, this means the machine is out of coins of denomination v. All queries after this point cannot use this denomination. It is guaranteed that each `X` event corresponds to a denomination v which the machine currently has in stock.

Output

Output k lines, where k is the number of query events ($c = \texttt{Q}$). On the i^{th} line, output the fewest number of coins needed to give change for the i^{th} query, or -1 if this is impossible.

Sample Input 1	Sample Output 1
4 7 1 2 5 10 Q 23 X 1 Q 23 X 5 Q 23 X 10 Q 22	4 6 -1 11

B: Big Integers

Time Limit: 2 seconds, Memory limit: 2G

Nick is preparing a problem for a programming contest about comparing big integers. He has decided on the input format for the integers: They will be expressed in base 62, with 0 through 9 representing digit values 0 through 9, lowercase letters `a` through `z` representing digit values 10 through 35, and uppercase letters `A` through `Z` representing digit values 36 through 61. For example, the string `Aa` would represent $36 \times 62 + 10 = 2242$.

The problem is to take two strings representing two distinct base 62 integers and determine which of the two is smaller. However, Nick wrote his judge solution incorrectly, assuming that the lexicographically smaller string is always the smaller integer.

Given some test cases, determine for each if Nick's solution would report the correct result.

Input

The first line of input contains a single integer t ($1 \leq t \leq 10^5$). This is the number of test cases.

Each test case consists of two lines.

The first line contains a single alphanumeric string of length at most 10^5.

The second line contains a single alphanumeric string of length at most 10^5.

Both strings are guaranteed to contain no unnecessary leading zeroes, and the two strings are guaranteed to be distinct.

The sum of the lengths of all input strings across all t test cases is guaranteed to be at most 2×10^6.

Output

For each test case, output a single line with `YES` if the lexicographically smaller string represents the smaller integer in base 62, and output a single line with `NO` otherwise.

Sample Input 1	Sample Output 1
2 icpc ICPC a bc	NO YES

This page is intentionally left blank.

C: Champernowne Subsequence

Time Limit: 2 seconds, Memory limit: 2G

The k^{th} Champernowne word is obtained by writing down the first k positive integers and concatenating them together. For example, the 10^{th} Champernowne word is 12345678910.

It can be proven that, for any finite string of digits, there exists some integer k such that the finite string of digits will appear as a subsequence in the k^{th} Champernowne word.

String s is a subsequence of string t if it is possible to delete some (possibly zero) characters from t to get s.

Given a string of digits, compute the smallest integer k such that the given string of digits is a subsequence of the k^{th} Champernowne word.

Input

The first line of input contains a single integer n $(1 \leq n \leq 10^5)$, the length of the string of digits.

The second line of input contains a string of n digits.

Output

Output a single integer k, the minimum integer such that the given string is a subsequence of the k^{th} Champernowne word.

Sample Input 1	Sample Output 1
2 90	10

Sample Input 2	Sample Output 2
2 00	20

This page is intentionally left blank.

D: Connecting Computers

Time Limit: 3 seconds, Memory limit: 2G

The tech team is gearing up for the PacNW Regional! There are n computers that need to be connected together for the contest, isolated from the outside internet. There are m bidirectional connections between computers, each connection requiring one of k different types of cable (for example, CAT5, RS232, MIDI, etc.) Using all the possible cable types, every computer is reachable from every other computer by following some set of cables. In addition, each pair of computers participates in at most one connection. Finally, to minimize leakage, no two distinct simple cycles of connected computers can have more than one computer in common (a simple cycle is a cycle where each computer can appear at most once, and two cycles are distinct if there is at least one connection present in one but not the other).

The tech team needs to make sure that every computer can communicate with every other computer, but doesn't want to use all the cable types if they don't have to. Can you help them figure out the answers to two questions: what is the minimum number of cable types they need to connect all of the computers, and how many subsets of cable types would allow every computer to communicate with every other computer?

Input

The first line contains three integers n, m, and k ($1 \leq n \leq 2 \cdot 10^5$, $n - 1 \leq m \leq 3 \cdot 10^5$, $1 \leq k \leq 24$) — the number of computers, the number of connections, and the number of cable types respectively.

The next line contains k strings: the i^{th} string is the name of the i^{th} cable type. Each cable name is made up of between 1 and 10 alphanumeric characters. It is guaranteed that the cable names are distinct.

The next m lines describe the connections between the computers. The i^{th} line contains two integers x and y ($1 \leq x < y \leq n$) and a string s (guaranteed to be one of the cable types).

Output

Output two lines, each containing a single integer. The first line should contain the minimum number of cable types the tech team needs connect all the computers. The second line should contain the number of subsets of cable types such that only installing those types would connect all of the computers.

Sample Input 1	Sample Output 1
6 7 4 CAT5 RS232 MIDI USBC 1 2 CAT5 2 3 MIDI 3 4 MIDI 2 4 CAT5 4 5 MIDI 5 6 MIDI 4 6 RS232	2 4

E: Exact Change

Time Limit: 1 second, Memory limit: 2G

While in Binaria, you find a store where you want to buy some presents for your friends. In Binaria, the currency is bits, and the coin denominations are the set of all integer powers of 2. You know that you want to spend at least a bits here, but no more than b bits.

When you make a purchase, you must pay with exact change. You have an unlimited number of bits that you can access from your bank account, but you can choose to withdraw them in whatever denominations you find most convenient. Carrying many coins is inconvenient though, so you wish to minimize the number of coins you carry with you.

Compute the minimum number of coins you need to bring with you such that you can pay any integer amount of bits between a and b, inclusive.

Input

The first line of input contains a single integer a, $1 \leq a < 2^{1000000}$. a will be written in base 2 with no leading zeros.

The second line of input contains a single integer b, $a \leq b < 2^{1000000}$. b will be written in base 2 with no leading zeros.

Output

Output a single integer k, the minimum number of coins you need to bring.

Sample Input 1	Sample Output 1
10101 101010	6

Sample Input 2	Sample Output 2
100 101	2

This page is intentionally left blank.

F: Free Solo

Time Limit: 5 seconds, Memory limit: 2G

Alex is a climber who is planning his next free solo expedition. He wants to do a completely new climb this time, one that nobody else has done before. Before he starts the climb, he wants to get a rough idea of what the shortest path to the top is.

In climbing, Alex uses his hands and feet to hold on to "holds," which are points on the wall he is climbing where he can put either his hands or feet to stabilize himself.

In this simplified simulation of the route, Alex is modelled as a single point, from which 4 limbs of up to length 1 extend out from. Holds on the wall are also single points, and can be used by any of his limbs, but one hold can only be used by one limb at a time. In order to stay safe, Alex must ensure that at least 3 of his limbs are attached to holds at all times during the climb.

Alex starts at coordinate $(0, 0)$ on 3 holds located at $(-0.2, 0)$, $(0, 0)$, and $(0.2, 0)$ and has a target hold at (T_x, T_y) he is trying to reach. Find the length of the shortest path traced out by Alex's location until he is able to put one limb on the target hold. Alex's location is allowed to be anywhere on the wall as long as he stays safe, including locations with negative y-coordinates.

Guarantees:

- No two holds will be within 10^{-3} units of each other.
- No two holds will be within 10^{-3} units of being distance 2 apart from each other.
- There will not exist a location on the wall where Alex is able to reach more than 6 holds.
- If Alex's reach increases or decreases by up to 10^{-6}, the length of the shortest path will not change by more than 10^{-5}.

Input

The first line contains an integer n $(1 \leq n \leq 30)$, the number of holds.

The next n lines contain two floating point numbers, the x and y coordinates of each hold $(-10 \leq x \leq 10, 0 \leq y \leq 10)$.

Floating point numbers will be expressed to exactly 5 decimal places.

The three starting holds are not included in the input. The target hold is the last hold given.

Output

Output the length of the shortest safe path to the target hold, or -1 if no such path exists. Your answer will be accepted if the absolute or relative error is within 10^{-4} of the judge's answer.

Sample Input 1	Sample Output 1
1 0.00000 1.50000	0.5000000000

Sample Input 2	Sample Output 2
1 0.00000 2.50000	-1

Sample Input 3	Sample Output 3
4 0.00000 0.50000 -0.80000 1.50000 0.80000 1.50000 0.70000 2.20000	1.3647093219

G: GCD Pairs

Time Limit: 5 seconds, Memory limit: 2G

In the Shape Galaxy, where all shapes are sentient beings, there is currently a feud between circles and squares. Circles want all pathways to be flat while squares argue that they should be evenly spaced inverted catenary shaped bumps. Because of this feud, circles have begun to dislike all square-biased numbers. A number s is square-biased if it is divisible by x^2, for some integer $x > 1$.

Mr. Circle has taken this feud to heart. He is given the assignment of calculating the greatest common divisor between all pairs of numbers in an array. He wants to go one step further and count the number of greatest common divisors that are not square biased.

Input

The first line of input contains a single integer, n $(1 \leq n \leq 10^5)$, representing the length of the array of numbers.

The next n lines contain the integers a_i $(1 \leq a_i \leq 10^{12})$ which comprise the numbers in the array.

Output

Output a single integer, the number of pairs (i, j) $(1 \leq i < j \leq n)$ such that $\gcd(a_i, a_j)$ is not square-biased.

Sample Input 1	Sample Output 1
6 3 4 6 12 4 1	12

This page is intentionally left blank.

H: Homework Help

Time Limit: 1 second, Memory limit: 2G

Alice was recently given a homework assignment to write a program that would output the number of inversions in any subarray of a given permutation. She happily turned it in and got full marks on her assignment. The next week, their homework assignment was to find the length of the longest increasing subsequences in the same array. Unfortunately, Alice had already thrown away the paper that contained the permutation she needed.

Luckily, this permutation was stored in the program she wrote. Unfortunately, she is now only able to query for the number of inversions in the subarrays of the permutation. As class is close to starting, she asks you for help to solve this problem in a timely manner.

A sequence a is a subsequence of an array b if it is possible to delete some (possibly zero) elements from b to get a. A sequence is increasing if every element is strictly greater than all preceding ones, and the LLIS of an array is the length of the longest increasing subsequence.

Interaction

This is an interactive problem.

Interaction starts by reading a single integer n ($1 \leq n \leq 10^3$), the length of the permutation p.

You are then able to make at most n queries of the form `? l r` where $1 \leq l \leq r \leq n$. Each query should be on a single line. After each query, read in a single integer, the number of inversions in the subarray $[l, r]$. An inversion is a pair of integers (x, y) where $l \leq x < y \leq r$ and $p_x > p_y$.

When you have determined the LLIS, output the answer in the form: `! ans` on a single line, where `ans` is the LLIS. After outputting the answer, your program should exit. If you attempt to read a response after outputting an answer, you will receive an arbitrary verdict.

Do not forget to flush the output after each query you output.

The interactor is not adaptive: When the interaction begins, the permutation p is already determined. It is guaranteed that each integer from 1 to n appears exactly once.

If the interactor receives any invalid or unexpected input, the interactor will output -1 and then immediately terminate. Your program should cleanly exit in order to receive a Wrong Answer verdict, otherwise the verdict that it receives may be an arbitrary verdict indicating that your submission is incorrect.

If your program terminates before outputting the answer, your submission will receive an arbitrary verdict indicating that your submission is incorrect.

You are provided with a command-line tool for local testing. The tool has comments at the top to explain its use.

Read	Sample Interaction 1	Write
3		
		? 1 3
1		
		? 1 2
1		
		! 2

I: Intuitive Elements

Time Limit: 1 second, Memory limit: 2G

Brandon is learning the periodic table! However, he doesn't like some of the elements because the symbol of the element contains letters which are not present in the name of the element. He finds this to be unintuitive, especially because in other contexts, he expects abbreviations to not introduce random letters.

Given a string and a proposed abbreviation, determine if Brandon would find it intuitive. Brandon finds an abbreviation intuitive if and only if every letter that appears in the abbreviation appears in the original string. Brandon does not look at the abbreviation carefully, so it is acceptable for a letter to appear more times in the abbreviation than in the original string, or for the letters to appear in a different order between the string and the abbreviation.

Input

The first line of input contains a single integer t ($1 \leq t \leq 10^3$). This is the number of test cases.

Each test case is represented on two lines.

The first line of each test case contains a single string a of length at least two and at most 50. This string only contains lowercase letters. The second line of the test case contains a single string b that is strictly shorter than a and also only contains lowercase letters.

Output

Output t lines, one for each test case.

For each test case, if all the letters in b appear in a, output `YES`. Otherwise, output `NO`.

Sample Input 1	Sample Output 1
4 magnesium mg silver ag aabb bbb aabb ba	YES NO YES YES

This page is intentionally left blank.

J: Rainbow Bowl Ranges

Time Limit: 2 seconds, Memory limit: 2G

You have a set of n bowls, arranged in a circle.

You have many balls of various colors. There are m different colors, and you have c_i balls of the i^{th} color.

You want to distribute all the balls into the bowls. To do this, for each color, you choose a contiguous range of bowls of size c_i and place one ball of that color in each bowl in the range. A contiguous range of bowls is a set of consecutive bowls around the circle. Ranges from different colors are allowed to overlap.

A bowl is *rainbow* if it contains one ball of each color. A *rainbow bowl range* is a contiguous range of rainbow bowls that cannot be extended by including another rainbow bowl.

You want to arrange balls in bowls to maximize the number of rainbow ranges.

Given the number of bowls and the number of balls of each color, what is the maximum number of rainbow bowl ranges that can be formed?

Input

The first line contains two integers, n $(2 \leq n \leq 10^9)$, m $(1 \leq m \leq 10^5)$.

The next m lines each contain a single integer, c_i $(1 \leq c_i \leq n)$.

Output

Print a single integer, the maximum number of rainbow bowl ranges that can be formed.

Sample Input 1	Sample Output 1
4 2 3 3	2

Sample Input 2	Sample Output 2
10 11 3 1 4 1 5 9 2 6 5 3 5	1

K: Sleeping on the Train

Time Limit: 2 seconds, Memory limit: 2G

Antonio is sightseeing in Line Town. Part of his sightseeing involves taking the famous Line Train. The Line Train goes through n stops conveniently numbered from 1 to n. The path the Line Train takes involves starting at stop 1, then going to every stop in numerically increasing order until it reaches stop n, at which point it turns around and goes to every stop in numerically decreasing order until it reaches stop 1, where it turns around and repeats its journey. When the train gets to either stop 1 or stop n, it lets all passengers that want to disembark leave the train. It then turns around, and then allows new passengers to board before heading to the next stop.

Antonio is traveling from stop a to stop b. Antonio is very sleepy, so he is not paying attention when he boards the train and could board a train initially heading in the wrong direction. Immediately upon boarding the train, he falls asleep and wakes up t times during the trip. Each time he wakes up, he notices that he is somewhere between stop s_i and $s_i + 1$. Since he is very sleepy, he does not know which direction the train is traveling in. Also, since he is not presently at his destination, he immediately falls back asleep.

After the t^{th} time waking up, Antonio decides he should stay awake for the rest of the trip. He stays on the train until the next time it stops at stop b, at which point he disembarks.

Compute the minimum number of times the train turned around while he was on it.

Input

The first line contains four integers, n $(2 \leq n \leq 10^9)$, t $(1 \leq t \leq 10^5)$, a, and b $(1 \leq a, b \leq n, a \neq b)$.

The second line contains t integers. The i^{th} integer, s_i $(1 \leq s_i < n)$, indicates that when Antonio woke up for the i^{th} time, he was somewhere between stops s_i and $s_i + 1$.

Output

Output the minimum number of times the train turned around while he was on it.

Sample Input 1	Sample Output 1
10 1 5 3 4	0

Sample Input 2	Sample Output 2
10 2 5 3 5 4	1

L: Training, Round 3

Time Limit: 1 second, Memory limit: 2G

Ashley is training for another programming contest on Brandon's Online Judge.

Ashley has k weeks left to train for her next programming contest. Her coach, Tom, is very busy and is no longer curating specific problems for Ashley to train on. At the start of every week, Tom picks p distinct problems independently and uniformly at random from the bank of n problems that Brandon's Online Judge has and assigns them for Ashley to work on. Tom generates a total of k sets of problems in this manner.

Ashley diligently solves every problem that Tom picks out. However, she gets annoyed if there exist two different weeks that share at least one problem in common, because she wants to solve unique problems.

Compute the probability that Ashley becomes annoyed.

Input

The first and only line of input contains three integers, n $(1 \leq n \leq 10^7)$, k $(1 \leq k \leq 10^7)$, and p $(1 \leq p \leq n)$.

Output

Let p be the probability that Ashley becomes annoyed. It can be shown that p can be written as a rational number $\frac{x}{y}$ with $\gcd(x, y) = 1$ and $y \not\equiv 0 \pmod{998244353}$. Define r to be the unique integer in $[0, 998244353)$ such that $r \cdot y \equiv x \pmod{998244353}$. Output r.

It can be shown that, under the constraints provided, r is guaranteed to exist and also be unique.

Sample Explanation

In the first sample, we can show that the probability Ashley becomes annoyed is $\frac{7}{9}$. Note that $110916040 \cdot 9 \equiv 7 \pmod{998244353}$, therefore the output for that test case is 110916040.

In the second sample, we can show that Ashley always becomes annoyed. Note that $1 \cdot 1 \equiv 1 \pmod{998244353}$, therefore the output for that test case is 1.

In the third sample, we can show that Ashley never becomes annoyed. Note that $0 \cdot 1 \equiv 0 \pmod{998244353}$, therefore the output for that test case is 0.

Sample Input 1	Sample Output 1
`3 3 1`	`110916040`

Sample Input 2	Sample Output 2
`3 2 2`	`1`

Sample Input 3	Sample Output 3
`3 1 1`	`0`

M: Word Game

Time Limit: 2 seconds, Memory limit: 2G

On the Word Game show, Ashley has selected n words and asks Brandon to combine them. Two words s and t can be combined if s has a suffix of length $k > 0$ that is a prefix of t. The result of combining them is a new word made of s concatenated with the last $|t| - k$ letters of t. If there are multiple values of k that are valid, any can be chosen.

Brandon must repeatedly take a pair of words from the list of words that can be combined, and replace them in the list with the combined word, until the list contains only a single word, and that word is as short as possible. If multiple final words of the same length are possible, Brandon must find the lexicographically first one.

Input

The first line of the input contains a single integer n $(1 \leq n \leq 5)$, the number of words to start out with.

The next n lines each contain a single word in lowercase letters of length at most 5.

Output

Output the lexicographically first word of minimum length Brandon can come up with. If it is not possible to come up with a single word, output -1.

Sample Input 1	Sample Output 1
2 aba bab	abab

Sample Input 2	Sample Output 2
3 ab bc ca	abca

Sample Input 3	Sample Output 3
2 x y	-1