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 2
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: 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.

B: Bus Assignment

Time Limit: 2 seconds, Memory limit: 2G

The Institution for Carrying People Carefully is responsible for managing the famous Line Bus in Line Town. The Line Bus goes through n stops conveniently numbered from 1 to n. At stop i, a_i people first get off the bus. Then, b_i people get on the bus. The bus starts out empty at stop 1 and then goes through the stops in numerically increasing order, eventually stopping at stop n where the bus empties.

When someone rides the Line Bus, they must be seated. A bus with capacity c has exactly c seats for passengers. Each rider of the Line Bus occupies exactly one seat. The driver of the Line Bus is not counted. The Institution for Carrying People Carefully wants to know what is the minimum capacity bus needed to run the Line Bus.

Input

The first line contains a single integer, n $(2 \leq n \leq 2 \cdot 10^5)$.

Each of the next n lines contains two integers, a_i and b_i $(0 \leq a_i, b_i \leq 10^9)$. It is guaranteed that at least one person boards the bus, at most 10^9 people board the bus over all stops, and that the bus will empty at stop n.

Output

Output a single integer, the minimum capacity bus needed to run the Line Bus.

Sample Input 1	Sample Output 1
4 0 3 1 2 2 1 3 0	4

C: Generalized FizzBuzz

Time Limit: 1 second, Memory limit: 2G

FizzBuzz is a common coding interview problem. The problem is as follows:

Given a positive integer n, for all integers i from 1 to n, inclusive:

- *If i is divisible by both 3 and 5, print "FizzBuzz".*
- *Otherwise, if i is divisible by 3, print "Fizz".*
- *Otherwise, if i is divisible by 5, print "Buzz".*
- *Otherwise, print i.*

We are interested in a generalized version of FizzBuzz:

Given three positive integers n, a, and b, for all integers i from 1 to n, inclusive:

- *If i is divisible by both a and b, print "FizzBuzz".*
- *Otherwise, if i is divisible by a, print "Fizz".*
- *Otherwise, if i is divisible by b, print "Buzz".*
- *Otherwise, print i.*

Given n, a and b, how many times are "Fizz", "Buzz", and "FizzBuzz" printed for a correct implementation?

Input

The first and only line of input contains three positive integers n, a and b $(1 \leq n, a, b \leq 10^6.)$

Output

Output three integers: the number of times "Fizz" is printed, the number of times "Buzz" is printed, and the number of times "FizzBuzz" is printed.

Sample Input 1	Sample Output 1
`17 3 5`	`4 2 1`

Sample Input 2	Sample Output 2
`10 3 3`	`0 0 3`

This page is intentionally left blank.

D: 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 ($1 \le t \le 10^3$). 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.

E: Mouse Pursuit

Time Limit: 3 seconds, Memory limit: 2G

Brandon is playing the newest idle game, Mouse Pursuit! The goal of this game is to pursue mice for cheese and glory.

In Mouse Pursuit, an *event* consists of pursuing a mouse. If the mouse is caught, the player might earn cheese and glory. However, if the mouse is not caught, the player might lose cheese and glory.

Given Brandon's recent events, Brandon wants to know how much cheese and glory he earned in the last k seconds.

Input

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

The next n lines take one of two forms:

- CAUGHT s c g: A mouse was caught exactly s seconds ago. The player gained c pieces of cheese and g units of glory.
- MISS s c g: A mouse was missed exactly s seconds ago. The player lost c pieces of cheese and g units of glory.

For all events, $0 \leq c, g \leq 10^6$ and $1 \leq s \leq 10^9$. It is guaranteed that no two events happened at exactly the same time.

The last line contains a single integer k. It is guaranteed that no event happened exactly k seconds ago.

Output

Output two integers - the number of pieces of cheese Brandon gained in the last k seconds, and the number of units of glory Brandon gained in the last k seconds.

Sample Input 1

```
3
CAUGHT 1 6 5
MISS 4 1 2
CAUGHT 8 0 3
5
```

Sample Output 1

```
5 3
```

This page is intentionally left blank.

F: Ruffians

Time Limit: 1 second, Memory limit: 2G

Ashley and Brandon are playing the new hit card game, Ruffians!

In Ruffians, ten cards are dealt out in a grid of two rows and five columns. Each card has a number on it from 1 to 9. Ashley and Brandon are both looking for a pair of cards that have the same number.

After playing this game for a while, they realize that there is always a pair of cards that have the same number. To make the game harder, they require that they find a pair of cards with the same number, and that the two cards are in different rows and different columns.

Given an arrangement of cards, determine if such a pair exists or not.

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 five integers, each between 1 and 9. The second line of the test case also contains five integers, each between 1 and 9. These two lines combined form the grid of two rows and five columns of cards.

Output

Output t lines, one for each test case.

For each test case, if there exists a pair of cards with the same number in different rows and different columns, output `YES`. Otherwise, output `NO`.

Sample Input 1

```
3
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
2 6 7 8 9
5 5 5 5 5
5 5 5 5 5
```

Sample Output 1

```
NO
YES
YES
```

This page is intentionally left blank.

G: 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`

H: 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.

I: 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.

J: 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`

K: 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 $(2 \le n \le 10^9)$, m $(1 \le m \le 10^5)$.

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

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

L: Taking Out the Trash

Time Limit: 3 seconds, Memory limit: 2G

Peter has way too much trash and he needs to take it all out.

Specifically, there are n bags of trash each with a specific weight. Peter can hold either one or two bags of trash per trip, and he can carry a maximum total of m milligrams of trash in a single trip. What is the minimum number of trips Peter needs to take to take out all the trash?

Input

The input starts with two integers n $(1 \le n \le 5 \cdot 10^5)$ and m $(1 \le m \le 10^9)$, the number of bags of trash and the maximum weight of trash Peter can carry.

The next line contains n integers, w_i $(1 \le w_i \le m)$, the weight of each bag of trash in milligrams.

Output

Output the minimum number of trips Peter needs to make to take out all the trash.

Sample Input 1	Sample Output 1
4 1000 100 900 200 900	3

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

This page is intentionally left blank.

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