Problem A. EyeQueue-Priority

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Pear company is preparing for presenting their new device called EyeWatch. This time they decided to change the rules of queuing customers. New rules are much easier:

Customers may purchase special priority certificates before start of sales. The cost of certificate is not fixed — customers are free to pay any amount of money they want. This certificate gives the customer a "priority"that is numerically equal to amount of money he payed for the certificate.

In the queue the next customer who should enter the shop is the one with largest priority among all people currently present in the queue, if there are several customers with such priority — one who came first is chosen.

Customers without certificates are considered to have priority 0.

Company is now just testing their idea, so, this system will be tested in only one store. Write a program which will simulate the queueing process in one store.

Input

The first line of input file contains the only integer number: $0 \leq N \leq 100000$ – the number of events your program should process.

The following $0 \leq N \leq 100000$ lines contain events your program need to process.

The line defining i-th event starts with symbol $+$ or $-$ which denotes type of event. $+$ means that new customer entered the queue. $-$ means that next customer with maximum priority (among all with equal priority — one who entered the queue first) enters the shop. Then, for $+$ event goes two integer numbers divided by space character : $0 \leq id_i < N, 0 \leq p_i < 10^9 - id$ (just number of customer, starting from 0) and priority of i-th customer.

Output

For each event of type $-$ print id of customer who enters the shop during this event on separate line.

Examples

standard input	standard output
6 + 0 10 + 1 5 + 2 7 - - -	0 2 1
10 + 0 0 + 1 0 + 2 0 - + 3 100 + 4 1 - - - -	0 3 4 1 2
12 + 0 0 + 1 0 + 2 0 + 3 1 + 4 1 + 5 1 - - - - - -	3 4 5 0 1 2

Problem B. Circular Shift

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given two strings s_1 and s_2. Check if s_2 is a right circular shift of s_1. If yes — print minimum non-negative K such that $CircShift(s1, K) = s_2$. If no — print -1.

$b = CircShift(a, 1)$ just moves all elements to the right and replaces first element with the last one:

$$b[0] = a[-1], \quad b[1:] = a[:-1]$$

$$e.g.: CircShift('abcde', 1) =' eabcd'$$

$CircShift(s2, K)$ just repeats this operation K times:

$$CircShift(s2, K) = CircShift(CircShift(s2, 1), K - 1)$$

Input

Input file contains two lines — strings s_1 and s_2 terminated by line break character. Both strings contain latin letters only: (a-z, A-Z). Length of both strings do not exceed $100000 : 0 < |s_1| \leq 100000, 0 < |s_2| \leq 100000$.

Output

Print single integer number $K = min\{k \geq 0 : CircShift(s_2, k) = s_2\}$, or -1 if K doesn't exist.

Examples

standard input	standard output
`abcde` `eabcd`	`1`
`abcde` `abcda`	`-1`
`abcdefgh` `fghabcde`	`3`