Selection Contest for ACM-HK Programming Contest 2018

The University of Hong Kong

May 5, 2018

DO NOT OPEN THIS PAPER UNTIL YOU ARE TOLD TO DO SO.

Sign in to the System

Please visit `http://147.8.177.161/pc2.html` and follow the instructions.

Important Notes

- The contest begins at **20:00** and lasts for **120 minutes**, ending at **22:00**.
- The problem set consists of **5 problems**.
- Printed and written notes are allowed, while internet access is prohibited.
- Please use **standard input/output** (e.g. `scanf`/`printf`, `cin`/`cout` or `System.in`/`System.out.print`) and do not print anything other than those required in the problem.
- For each problem, the running time limit is **4 seconds** per test case.
- The real time scoreboard is available at `http://147.8.177.161/board.html`.

Problem A – Complete Naebbirac's sequence

Author: Yonny Mondelo Hernández, Cuba

Naebbirac is a young and easy-to-get-bored sailor. He likes sequences of integers and to come up with ways to classify them. Naebbirac says that a sequence is *complete* for a chosen integer K, if the sequence only contains integers between 1 and K, and each integer between 1 and K appears the same number of times.

Based on that, Naebbirac created a game to entertain himself and his peers, when the waters calm down and there's not much they can do to spend their time in the middle of the ocean.

First he chooses a positive integer K and then he uses chalk to draw on the deck a sequence S having N integers between 1 and K. After that he challenges one of his peers. The goal of the challenged peer is to turn the sequence S into a *complete* sequence by performing exactly one of the following three possible operations:

- "`-x`": remove one occurrence of integer x from S;
- "`+x`": add a new integer with value x in S; or
- "`-x +y`": replace one occurrence of integer x from S by an integer with value y.

Naebbirac is quite smart. He never writes a sequence that is already *complete* and often the written integers don't follow a pattern, making it quite hard to find an operation that solves the puzzle. One of your friends, that usually sails with Naebbirac, is tired of always losing the game. Are you able to help your friend and create a computer program that can find a solution to Naebbirac's game before they go on their next trip?

Input

The first line contains two integers K ($3 \leq K \leq 1000$) and N ($1 \leq N \leq 10^4$), indicating respectively the integer that Naebbirac chooses at the beginning of the game, and the length of the sequence written on the deck. The second line contains N integers $S_1, S_2, \ldots, S_N$ ($1 \leq S_i \leq K$ for $i = 1, 2, \ldots, N$) representing the written sequence; you can safely assume that the sequence is not *complete*.

Output

Output a single line with the description of the operation that allows your friend to win the game or an "`*`" (asterisk) if there is no way to win. The description of the operation must follow the format shown on the statement, i.e. "`-x`", "`+x`" or "`-x +y`".

Sample input 1	**Sample output 1**
`3 5` `1 3 2 3 1`	`+2`

Sample input 2	**Sample output 2**
`3 7` `1 2 3 3 3 2 1`	`-3`

Sample input 3	**Sample output 3**
`3 6` `3 1 2 1 3 1`	`-1 +2`

Sample input 4	**Sample output 4**
`3 6` `2 3 2 2 2 1`	`*`

Problem B – Enigma

Author: Jeferson Lesbão, Brasil

The world famous pirate Cornelius "Cheesehead" Bakker was a renowned astronomer and mathematician. He buried most of his treasury in the Caribbean island of Saint Basil, where Pico Colombo is a well-known geographic reference mark. Cheesehead disappeared when his fleet of three ships was caught in a hurricane in 1617. Perhaps by some kind of premonition, before his fatal excursion he wrote in a letter to one of his nieces in the Netherlands the exact distance to his hidden treasure, from Pico Colombo in the south direction.

Wary that the map would end up in the wrong hands, Cheesehead used his math skills as an insurance against robbers. Instead of writing in the letter the number indicating the distance, he multiplied it by a second number N, and wrote the result D in the letter, together with the value of N and an explanation of the computation he had done. He knew that even if some unwanted person had the letter, he or she would have to know how to divide two numbers, which very few robbers could at that time. Unfortunately, when the letter arrived in Europe, Cheesehead's niece had joined a convent to become a nun and did not even bother to open the letter.

Exactly four centuries afterwards, Maria came into possession of a chest containing the belongings of her ancestor nun. And you can imagine her surprise when she found the letter, still unopened! Maria is planning an excursion to seek for Cheesehead's treasure, but she needs your help. Although the value of N is intact and she can read it, the number D has been partially eaten by moths so that some of its digits are unreadable. The only clue Maria has is that the leftmost digit of D is not zero because Cheesehead said so in the letter to his niece.

Given the partial representation of D and the value of N, you must determine the smallest possible value for D so that it is a multiple of N and does not have leading zeros.

Input

The input consists of a single line that contains a non-empty string S of at most 1000 characters and an integer N ($1 \leq N \leq 1000$). Each character of S is either a decimal digit or the character "`?`" (question mark); the leftmost character is not "`0`" and at least one character is "`?`".

Output

Output a single line with an integer D without leading zeros indicating the smallest multiple of N that has $|S|$ digits and such that the digits in S are coincident with the corresponding digits in D. If there exists no such an integer D, write an "`*`" (asterisk) to the output.

Sample input 1	**Sample output 1**
`1?????????????????????????????? 2`	`100000000000000000000000000000`

Sample input 2	**Sample output 2**
`??????????????????????????????1 2`	`*`

Sample input 3	**Sample output 3**
`?294?? 17`	`129404`

Problem C – Hard choice

Author: Inés Kereki, Uruguay

In long flights, airlines offer hot meals. Usually the flight attendants push carts containing the meals down along the aisles of the plane. When a cart reaches your row, you are asked right away: "Chicken, beef, or pasta?" You know your choices, but you have only a few seconds to choose and you don't know how your choice will look like because your neighbor hasn't opened his wrap yet...

The flight attendant in this flight decided to change the procedure. First she will ask all passengers what choice of meal they would prefer, and then she will check if the number of meals available in this flight for each choice are enough.

As an example, consider that the available number of meals for chicken, beef and pasta are respectively $(80, 20, 40)$, while the number of passenger's choices for chicken, beef and pasta are respectively $(45, 23, 48)$. In this case, eleven people will surely not receive their selection for a meal, since three passengers who wanted beef and eight passengers who wanted pasta cannot be pleased.

Given the quantity of meals available for each choice and the number of meals requested for each choice, could you please help the flight attendant to determine how many passengers will surely not receive their selection for a meal?

Input

The first line contains three integers C_a, B_a and P_a ($0 \leq C_a, B_a, P_a \leq 100$), representing respectively the number of meals available for chicken, beef and pasta. The second line contains three integers C_r, B_r and P_r ($0 \leq C_r, B_r, P_r \leq 100$), indicating respectively the number of meals requested for chicken, beef and pasta.

Output

Output a single line with an integer representing the number of passengers that will surely not receive their selection for a meal.

Sample input 1	**Sample output 1**
`80 20 40` `45 23 48`	`11`

Sample input 2	**Sample output 2**
`0 0 0` `100 100 100`	`300`

Sample input 3	**Sample output 3**
`41 42 43` `41 42 43`	`0`

Problem D – Imperial roads

Author: Edwin Niño, Colombia

The roads of Cubiconia are in a dire state, after years of neglect and lack of maintenance. Each road connects two different cities A and B and can be traveled in both ways (from A to B or from B to A). There is at most one road between each pair of cities, and using the existing roads it is possible to travel between any pair of cities. The new emperor of Cubiconia has just raised the taxes (again!), but promised to repair at least some of the roads, guaranteeing that Cubiconians will be able to travel between any pair of cities using only restored roads.

The Department of Public Works have calculated the cost of repairing each individual road. Now they want to calculate the minimum cost for repairing a set of roads so that the emperor's promise is made true. This is not easy because the emperor wants the set of repaired roads to include one particular road, but he has not yet decided which particular road to include: could be the one that connects the city where his castle is to the city where his daughter's royal residence is, or the road that connects the city where his summer palace is to the only city by the seaside, or... Fearing the emperor will take too long to decide, the engineers want your help.

Given the description of the roads in Cubiconia, with their respective repairing costs, you must write a program to answer a set of queries. For each query you will be given one specific road that should be repaired, and must determine the minimum cost for repairing a set of roads (including the given specific road) so that Cubiconians will be able to travel between any pair of cities using only restored roads.

Input

The first line contains two integers N ($2 \leq N \leq 10^5$) and R ($N - 1 \leq R \leq 2 \times 10^5$), representing respectively the number of cities and the number of roads in Cubiconia. Cities are identified by distinct integers from 1 to N. Each of the next R lines describes a road with three integers A, B ($1 \leq A < B \leq N$) and C ($1 \leq C \leq 10^4$), indicating that there is a road between cities A and B and the cost of repairing it is C. There is at most one road between each pair of cities, and using the existing roads it is possible to travel between any pair of cities. The next line contains an integer Q ($1 \leq Q \leq 10^5$) representing the number of queries. Each of the next Q lines describes a query with two integers U and V ($1 \leq U < V \leq N$), indicating the specific road that should be repaired. There are no repeated queries.

Output

Output Q lines, each line with an integer indicating the answer to the corresponding query of the input, that is, the minimum cost for repairing a set of roads (including the specific road in the query) so that Cubiconians will be able to travel between any pair of cities using only restored roads.

Sample input 1	**Sample output 1**
3 3 1 2 10 2 3 5 1 3 7 3 2 3 1 2 1 3	12 15 12

Sample input 2	**Sample output 2**
4 4 1 2 1 2 4 1 2 3 100 1 4 50 1 1 4	151

Sample input 3	Sample output 3
5 7 1 2 8 1 3 10 2 4 5 2 3 12 4 5 4 3 5 14 1 5 20 3 2 3 1 5 3 5	29 39 31

Problem E – Jumping Frog

Author: Gabriel Poesia, Brasil

Pog the Frog wants to compete in the World Frog Jump competition, which will take place in Nlogonia. In the competition, each frog must perform a sequence of acrobatic jumps in a specially built arena. The arena is composed of N equally spaced positions around a circumference (the arc between two adjacent positions is always the same length) where each position can be either a rock or a pond. The positions are numbered sequentially from 0 to $N-1$ in the clockwise direction, so that judges can easily make notes about which jumps were performed in each position. Thus, position 0 is adjacent to positions 1 and $N-1$ in the arena.

The competition rules stipulate that the sequence of jumps of each frog must start at a rock, always go from a rock to another rock, and finish at the same position it started. The rules do not require frogs to use every rock in the arena for their sequence of jumps.

Pog the Frog is currently practicing for the competition. He must develop two skills. First, he needs to get better at jumping from one rock to another, since landing on either a pond or outside of the marked positions can mean disqualification. Besides that, he must learn impressing acrobatic moves. With that in mind, he has decided on a practicing strategy. In the beginning of each practice session, Pog the Frog will pick a starting rock and an integer *jump length* K between 1 and $N-1$. After that, whenever he is standing on a rock numbered i, he will aim his next acrobatic jump at the rock whose number is obtained by getting the remainder of the division of $i+K$ by N. He will stop when he lands on the starting rock. For example, if the arena has 3 positions, all of them rocks, and Pog the Frog starts at position 0 and picks $K=2$, he will first jump from rock 0 to rock 2, then to rock 1, and finally jump back to rock 0. At this point, his practice session ends.

Given the description of the N positions in the arena, help Pog the Frog by answering this question: how many distinct values of K can he choose for his practice sessions, if he can use any rock as a starting position for his sequence of jumps?

Input

The input consists of a single line that contains a string S of N characters ($3 \leq N \leq 10^5$), representing the positions in the arena. The i-th character of S ($i = 0, 1, \ldots, N-1$) indicates that the position i in the arena is either a rock (uppercase letter "`R`") or a pond (uppercase letter "`P`").

Output

Output a single line with an integer representing the number of distinct jump lengths that Pog the Frog can choose for his practice sessions, given that he can use any rock as a starting position for his sequence of jumps.

Sample input 1	**Sample output 1**
`RRR`	`2`

Sample input 2	**Sample output 2**
`RRPR`	`1`

Sample input 3	**Sample output 3**
`PRP`	`0`