Problem A: Ducks in a Row

Filename: `ducks`
Time limit: `3 seconds`

Sri is playing a game with ducks, geese, and a magic wand. First he puts all his ducks in a row. Next his friend Srinivas inserts some geese between the ducks at different places. Sri can then use his magic wand to flip some of the ducks and geese.

Each use of his wand can be defined formally:

1) Sri can select some contiguous sequence of ducks and geese.
2) All birds that were ducks before using the wand are now geese.
3) All birds that were geese before using the wand are now ducks.

Sri has an objective to succeed at the game. He must turn the row into at least ***k*** maximal runs of consecutive ducks of length at least ***n***. A maximal run is a sequence of ducks that does not have a duck immediately to its left or right. For example, the following row of birds has 4 maximal runs of ducks of lengths 2, 3, 3, and 1, respectively:

D D G G G G D D D G D D D G D

Sri needs to find the minimum number of wand uses to meet his objective. There can be other maximal runs of consecutive ducks at the end of the game, maybe some of length <***n***, but there must be at least ***k*** of at least length ***n***.

Input

The first line of input will contain two integers ***n*** and ***k*** (1 ≤ ***n***, ***k*** ≤ 2,000), where ***n*** is the minimum length of each sequence of ducks that Sri desires, and ***k*** is the minimum number of sequences of ducks that Sri desires.

The second line will contain a single string, ***s*** (1 ≤ |***s***| ≤ 2,000), consisting of only the capital letters D and G. They represent the row of birds before Sri starts using his magic wand, where D is a duck and G is a goose.

Output

On a line by itself, output a single integer, the minimum number of times he must use his wand to meet his desired property or **-1** if it is not possible.

Samples

Input	Output
2 2 DDDGD	1
2 3 GGDGGDGG	1

Problem B: Hardest Problem

Filename: `hardest`
Time limit: `1 second`

Arup was worried somebody might solve the set for Contest 3, so he asked Spencer to make an extremely hard problem. Spencer made the problem, but when Jacob tried test-solving it, he was stumped. Spencer wondered, is this problem too hard for an SI contest? Probably, but it was too late to think of another problem.

Can you solve the problem Jacob couldn't? The problem is as follows:

Given an array ***A*** of length ***N*** we want to partition it into one or more contiguous groups. Once partitioned, we will take the XOR of the elements in each group. Finally, we will obtain the sum of these groups' XOR values. Bessie wants to partition the array such that this sum is minimal. Farmer John wants to partition the array such that this sum is maximal. What is the difference between their sums?

Input

The first line of input contains one positive integer: ***N*** ($1 \le N \le 10^5$). The second input line will contain ***N*** space separated positive integers less than 10^9, these are the elements of ***A*** (in order).

Output

Print out the difference between the sums Bessie and Farmer John calculate.

Samples

Input	Output
`4` `3 9 2 4`	`6`
`5` `1 2 4 8 16`	`0`

Problem C: Mad Mathematician

Filename: `madmath`
Time limit: `2 seconds`

Fatt Montaine, after a disgruntled childhood, has dedicated his life to finding interesting numbers. Currently he is working with properties of numbers surrounding the sum of the divisors of a number such as perfect, abundant, and weird numbers. For example the divisors of 12 are 1,2,3,4,6, and 12 so the sum of the divisors of 12 is 28. Today he has discovered an amazing number, but he forgot what it was. Thankfully he still has a way of getting the number back as he still has sum of the divisors of the number, which he denotes “S”. Fatt has been producing numbers by multiplying a set of smaller numbers together. However even if Fatt takes product of the set he wouldn’t be able to tell if it was the amazing number just by looking at it. He would need to recompute the sum of divisors of the product and compare it to S. However, computing the sum of divisors of the products of all these sets took him years! Being the sly fox that he is, Fatt knows that he can identify the correct set from obtaining just the remainder of the sum of the divisors of the product of the set when divided by $10^9 + 7$, Fatt’s favorite prime number.

Input

The first line of the input contains a single positive integer n ($1 \le n \le 400$), the number of integers in the set. The second line of input contains n integers a_0 - a_{n-1} ($1 \le a_i \le 500{,}000$). The product of these numbers may not necessarily fit into a 64-bit integer.

Output

Let the product of all a_i be denoted as P. Output sum of the divisors of P modulo $10^9 + 7$.

Samples

Input	Output
3 2 2 3	28
6 1000 2000 3000 4000 5000 6000	822202011

Problem D: As Easy As C-A-B

Filename: `cab`

Time limit: `1 second`

We all know how to alphabetize a list when you know the order of the alphabet. But can you find the order of the alphabet from an ordered list of words?

Consider the ordered list [cab, cda, ccc, badca]. It is clear that 'c' comes before 'b' in the underlying alphabet because 'cab' comes before 'badca'. Similarly, we know 'a' comes before 'd', because 'cab' < 'cda', 'a' comes before 'c' because 'cab' < 'ccc', and 'd' comes before 'c' because 'cda' < 'ccc'. The only ordering of these four letters that is possible is adcb.

Of course, it may not work out so well. If the word list is [abc, bca, cab, abc] there is no alphabet that works. The list is inconsistent. If the word list is [dea, cfb] we don't know about the relative positions of any of the letters other than ‘c’ and ‘d’. The list is incomplete. Every list will fall into exactly one of the following three categories:

1. The list is correct if a single alphabet will yield the ordering
2. The list is incomplete if more than one alphabet will yield the ordering
3. The list is inconsistent if no alphabet will yield the ordering

Given a list of words, determine if the list is correct, incomplete or inconsistent, and if it is correct, give the single underlying ordered alphabet.

Input

The first line of input contains a lowercase letter ***last***, and an integer ***n*** ($1 \le n \le 100$). Each of the following ***n*** lines will have a string ***s*** ($1 \le |s| \le 50$) consisting only of lowercase letters ‘a’-***last***.

Output

If the list is correct, and it is possible to uniquely determine the ordering of the letters ‘a’-last, output that ordering as a single string. If the list is incomplete, and there’s not enough information to determine the positions of all the letters, output 0 (zero). If the list is inconsistent in any way then output 1.

Samples

Input	Output
d 4 cab cda ccc badca	adcb
c 4 abc bca cab abc	1
f 2 dea cfb	0
b 3 a bb b	1

Sample Explanation:

The first three cases are described in the problem description and the last case is inconsistent because there is no alphabet for which bb comes before b.

Problem E: Charles is Always Late!

Filename: `charles`

Time limit: `1 second`

If there is one thing that everyone on Programming Team knows, it's that Charles is always late. Maybe he's on Miami time, but nobody knows for sure. Whatever the case, he's late to class, practice, and important TA meetings.

It turns out that depending on the path Charles chooses to take from his dorm to the Programming Team Lab, he might get sidetracked along the way by certain things, especially on Tuesday and Thursday afternoons in Spring semester, which is when the unofficial dog club meets on Memory Mall.

The UCF campus can be described as an undirected graph. Walkways are edges and junctions connecting sidewalks are nodes. For a normal person, each edge takes a certain amount of time to walk from one side to the other. But Charles might walk along that edge faster or slower.

Given the graph describing UCF's campus, output how much longer than normal it takes Charles to get from his dorm (at node ***1***) to the PTL (at node ***n***).

Input

The first line contains two integers, ***n*** ($1 \le n \le 2000$) and ***m*** ($1 \le m \le min(\binom{n}{2},\ 10^5)$), the number of junctions and walkways, respectively. Each of the next ***m*** lines contain three integers representing a walkway: ***u*** ($1 \le u \le n$), ***v***($1 \le v \le n$, $u \ne v$) and ***t*** ($1 \le t \le 10^5$), representing that intersections ***u*** and ***v*** are connected by a walkway that takes ***t*** minutes to traverse from one side to the other. Next is a line containing ***m*** integers, the i^{th} of which is c_i. Charles takes $t_i + c_i$ minutes to walk the i^{th} edge. It is guaranteed that no edge **i** exists such that $t_i \le 0$ or $t_i + c_i \le 0$.

Output

Output how much longer (or shorter) than normal it takes Charles to get to the PTL from his dorm. For example, if Charles is late, output a positive number representing how many minutes longer it takes him to get to the PTL. If he's early (unlikely), output a negative number representing how many fewer minutes it takes Charles to get to the PTL; if he's on time, print zero.

Samples

Input	Output
3 3 1 2 10 2 3 10 1 3 10 -5 0 15	5
3 3 1 2 10 2 3 10 1 3 10 5 0 0	0

Problem F: Fire Sale

Filename: `firesale`
Time limit: `2 seconds`

There is a fire sale going on at Anya's favorite toy store. As Arup wants to make Anya happy, he gives her a maximum allowance of k dollars. It is Anya's goal to buy as many toys as possible without spending more than k dollars.

However, the fire sale has certain rules. All of the toys are lined up in a row, and Anya can only buy toys that are in one contiguous sequence. Additionally, the cost of the sequence of toys is the cost of the most expensive toy minus the cost of the least expensive toy.

Input

The first line of input will contain two space separated integers, n ($1 \le n \le 10^6$), k ($1 \le k \le 10^6$), representing the number of toys in the store and the maximum amount of money Arup is willing to spend, respectively. The following line contains n space separated integers, a_i ($1 \le a_i \le 10^6$), the costs of each toy.

Output

On a single line, output the most toys Anya can buy without going over her allowance.

Samples

Input	Output
5 1 5 5 5 5 5	5
5 2 2 1 3 4 5	3
6 4 1 1 6 9 8 10	4