Problem A: Friday the 13^{th}

Filename: `friday`
Time limit: `3 seconds`

Today is Friday the 13^{th}, but UCF has nonetheless scheduled campus orientation for scores of unsuspecting freshmen. As you may know, UCF buildings become haunted on Friday the 13^{th}, cursing some of the people that pass through their halls.

The University's Paranormal Research Team (PRT) has determined that some of the M buildings are more haunted than others. Each building i has some haunting value h_i, a degree of danger in a unit of measurement known only to the team. In anticipation of the orientation tours, the PRT installed scanners designed to evaluate the degree of haunting an individual can withstand without becoming cursed. Using these scanners, they've determined how much haunting each of the N freshman at orientation can withstand without being cursed.

Now the PRT wants to determine the fallout of Friday the 13^{th} on this freshman class, assuming each student passes through every building once during orientation. For each student, find the number of buildings they will pass through which curse them.

Input

The first line of input will contain two integers N and M ($1 \le N, M \le 10^6$), the number of students at orientation and the number of buildings on the tour, respectively. The next line contains N integers f_i, indicating that freshman i will be cursed by buildings with a haunting value of at least f_i ($1 \le f_i \le 10^9$). The next line contains M integers h_i, indicating that building i has a haunting value of h_i. ($1 \le h_i \le 10^9$).

Output

Print N numbers, the i^{th} of which indicates the number of buildings which will curse student i, on a line by itself. Print a space between each number, but not after the last number.

Samples

Input	Output

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

`7 5`
`7 9 2 1 4 2 2`
`8 2 1 3 3`
	`1 0 4 5 1 4 4`

Problem B: Semiprime Life

Filename: `semiprime`
Time limit: `1 second`

Many people live charmed lives, but Arup prefers to live a semiprime life. A semiprime is an positive integer that is the product of precisely two not necessarily distinct prime numbers. For example 15 = 3 x 5, 49 = 7 x 7, and 143 = 11 x 13, are all semiprime numbers.

To live a semiprime life, each vacation you take must be X days long, where X is a semiprime number. Arup is considering several vacations, each lasting a different number of days. Help him determine how many of these vacations he'll be able to take.

Input

The first line of input will contain a single positive integer ***n*** ($1 \le n \le 50$), the number of vacations Arup is considering taking. The second line of input will contain ***n*** space separated distinct integers, each in between 2 and 1,000, representing the lengths of the potential vacations.

Output

On a single line, output the number of the potential vacations that are a semiprime number of days long.

Samples

Input	Output
`4` `15 22 18 49`	`3`
`5` `10 11 12 13 14`	`2`

Problem C: Zebra Zoo

Filename: `zoo`
Time limit: `2 seconds`

Zachariah the zookeeper has lost his zebras! The zebras have escaped from their habitat and are scattered throughout the zoo. Fortunately, the zebras are asleep right now, so right now is the perfect time to catch them. Zachariah has lent you a helicopter so that you can get a full view of the zoo and find his zebras for him.

Input

Each test case will begin with a single line containing two integers, ***R*** and ***C*** ($1 \le R, C \le 100$), representing the length and width of the zoo border, respectively. Following will be ***R*** lines containing ***C*** characters, each character being either a lowercase "`z`" or a "`.`". If the character is a "`z`", that means that there is a zebra located on that row and column.

Output

For each zebra in the zoo, print out a line containing the row and column where it is sleeping in the zoo. These lines should be in order of increasing row, with ties broken by column value, from smaller to larger. Zachariah is not crazy, so there is guaranteed to be at least one zebra in the zoo.

Samples

Input	Output

`3 4`
`..z.`
`z.zz`
`....`
	`1 3`
	`2 1`
	`2 3`
	`2 4`

`2 2`
`zz`
`zz`
	`1 1`
	`1 2`
	`2 1`
	`2 2`

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`