Problem A: Double Chomp

Filename: `chomp2`
Time limit: `2 seconds`

Nom Chomsky the chain chomp regularly sneaks bites of food from Bowser's dinner table. All the food on the table is arranged on a line and Chomsky is able to eat any contiguous section in a single bite, but he can only manage up to two bites before being caught. Chomsky knows how tasty each dish is and wishes to maximize the combined tastiness of his two bites. However he must be careful because some dishes taste so bad that they have a negative tasty value! Find the maximum tastiness Chomsky can obtain with two or fewer bites.

Input

The first line of input contains a single positive integer, n ($n \leq 10^5$), representing the number of food items on the table. The second line contains n space-separated positive integers. The i^{th} integer, t_i ($-1000 \leq t_i \leq 1000$), on this line represents the tastiness of the i^{th} food item from the left on the table.

Output

For each test case output the maximum sum of tastiness Nom Chomsky can enjoy if he takes two or fewer bits.

Samples

Input	Output
`1` `-1`	`0`
`2` `1 -2`	`1`
`11` `1 -9 5 -1 2 -3 1 -4 3 -4 1`	`9`

Solution for the third sample test case with bites bolded: `1 -9` **`5 -1 2`** `-3 1 -4` **`3`** `-4 1`

Problem B: Geography Assignment

Filename: geography
Time limit: `3 seconds`

Arnold is a student at the University of Carpathian Flamingoes. He is taking a geography class, and it's really hard. His teacher has asked him to determine which cities within a county can reach each other. However, some counties can have lots of cities, so Arnold wants you to write a program to help him find the answer. Also, there's a quirk of the country that the University is in: the roads only go in one direction!

Input

The first line of the input will contain two integers, ***n*** ($2 \le n \le 500$) and ***m*** ($1 \le m \le 1000$), the number of cities in the county and the number of roads in the county, respectively.

The next m lines will each describe a road.

Each line of these lines will contain two integers, ***a*** and ***b*** ($1 \le a, b \le n$, $a \neq b$) indicating that there is a road from city a to city b. There may or may not be a road from city b to city a. Each road listed will be unique.

Output

Output the number of pairs of cities (a, b), with a ≠ b, such that it's possible to travel from city a to city b, directly or indirectly. Note that if it's possible to travel from city a to city b AND it's possible to travel from city b to city a, then these should add 2 to the count.

Samples

Input	Output
5 4 1 3 3 2 4 5 5 4	5

Explanation of Sample Input 1:
City 1 can visit cities 2 and 3. City 3 can visit city 2. City 4 can visit city 5. City 5 can visit city 4.

Problem C: Grid

Filename: `grid`
Time limit: `2 seconds`

You are on an ***n***x***m*** grid where each square on the grid has a digit on it. From a given square that has digit ***k*** on it, a move consists of jumping exactly ***k*** squares in one of the four cardinal directions. A move cannot go beyond the edges of the grid; it does not wrap. What is the minimum number of moves required to get from the top-left corner to the bottom-right corner?

Input

The first line of input contains two space-separated integers ***n*** and ***m*** ($1 \le n, m \le 500$), indicating the size of the grid. It is guaranteed that at least one of ***n*** and ***m*** is greater than 1.

The next ***n*** lines will each consist of ***m*** digits, with no spaces, indicating the ***n***x***m*** grid. Each digit is between 0 and 9, inclusive.

The top-left corner of the grid will be the square corresponding to the first character in the first line of the test case. The bottom-right corner of the grid will be the square corresponding to the last character in the last line of the test case.

Output

Output a single integer on a line by itself representing the minimum number of moves required to get from the top-left corner of the grid to the bottom-right. If it isn't possible, output -1.

Samples

Input	Output

`2 2`
`11`
`11`
	`2`

`2 2`
`22`
`22`
	`-1`

`5 4`
`2120`
`1203`
`3113`
`1120`
	`6`

Problem D: Bodies of Water

Filename: `water`
Time limit: `1 second`

For those who don't like regular images, ASCII Maps Inc. has created maps that are fully printable ASCII characters. Each map is a rectangular grid of lowercase English letters, where each letter stands for various locations. In particular, 'w' stands for water and the other 25 letters represent various different land locations. For this problem, we are interested in counting the number of bodies of water on a given ASCII map. A body of water is a maximal set of contiguous grid squares on the ASCII map where each square in the body of water shares a boundary with at least one other square in the body of water. Thus, for two grid squares to be part of the same body of water, one must be above, below, to the left, or to the right of the other grid square.

Input

The first line of input consists of two space separated integers, ***r*** ($1 \le r \le 50$) and ***c*** ($1 \le c \le 50$), the number of rows and columns, respectively for the input map. The next ***r*** lines will each contain ***c*** lowercase English letters, representing the corresponding row of the input map.

Output

On a line by itself, output the number of bodies of water in the input map.

Samples

Input	Output

`5 6`
`waaaww`
`wawawc`
`bbbbwc`
`wwwwww`
`dddddd`
	`3`

`2 8`
`wxwxwxwx`
`xwxwxwxw`
	`8`

Problem E: All Your Bases

Filename: `bases`
Time limit: `1 second`

Jason is having trouble with his math homework. His professor asked him to convert a number from one base to another. Can you help him do this?

Input

The input consists of three positive integers, ***a***, ***b***, and ***c*** ($2 \le b, c \le 36$, $b \ne c$). It is guaranteed that all of the characters of ***a*** will be valid characters in base ***b*** and that the value in base 10 of ***a*** will be a positive integer one million or less. All letters in the input will be upper case letters.

Output

The output will be a single integer, ***a***, after being converted from base ***b*** to base ***c***.

Samples

Input	Output
`1010 2 8`	`12`

Input	Output
`ABC 16 5`	`41443`

Input	Output
`1022 3 12`	`2B`

Problem F: Gravity

Filename: `gravity`
Time limit: `1 second`

Consider a 2D grid, which contains apples, obstacles, and open spaces. Gravity will pull the apples straight down, until they hit an obstacle, or the bottom of the grid, or another apple which has already come to rest. Obstacles don't move. Given such a grid, determine where the apples eventually settle.

Input

The first line of input contains two integers, ***r*** and ***c*** (1 ≤ ***r***, ***c*** ≤ 100), which are the number of rows and the number of columns of the grid. On each of the next r lines will be c characters: 'o' (lowercase 'Oh') for an apple, '#' for an obstacle, and '.' for an open space.

Output

Output the grid, after the apples have fallen.

Samples

Input	**Output**
3 3 ooo #.. ..#	o.. #.o .o#
4 2 oo oo o. ..	.. o. oo oo

Problem G: Team Placements

Filename: `team`
Time limit: `1 seconds`

Dr. Orooji likes to keep track of how well the UCF Programming Team does at Southeast Regionals. Therefore, over the years, he counts how many times the team places first, second, and third. However, he only counts the best finish for that year. For example, if the team places first and second in the same year, he will only increase his count of how many times the team has finished first. Note that under ICPC rules, it is impossible for there to be a tie at any place.

Input

The first line of input will be a single integer n ($1 \le n \le 50$), the number of years the team has competed at Southeast Regionals. The next $2n$ lines will have the following format: the first line will contain a single integer m ($1 \le m \le 100$), the number of teams from UCF that competed. The next line will contain m integers. The i^{th} integer, p_i ($1 \le p_i \le 200$) is the place the i^{th} team from UCF placed. Each of the these m integers will be unique.

Output

The output will consist of three space separated integers, a, b, and c, the number of times the UCF Programming Team has finished first, second, and third, respectively, on a line by itself.

Samples

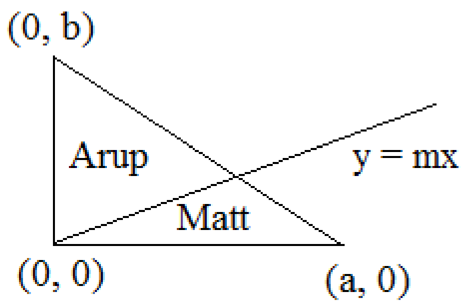
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>4
2
1 5
4
5 2 4 3
4
2 1 3 4
1
1</pre></td><td><pre>3 1 0</pre></td></tr>
<tr><td><pre>2
10
2 99 5 4 7 12 45 8 1 10
3
4 6 3</pre></td><td><pre>1 0 1</pre></td></tr>
</table>

Problem H: Arup's Triangles

Filename: `triangles`
Time limit: `1 second`

Arup has just been bequeathed a piece of land by a long lost uncle who knew of Arup's love of triangles. The piece of land is in the shape of a right triangle, which can be modeled on the Cartesian plane with legs on the positive x and y axes. It turns out that Arup lost a bet to Matt (over a programming team problem, of course), and the terms of that bet, as strange as they may seem, were that if Arup were to be given any item of value, he must share part of it with Matt. It just so turns out that Matt loves triangles as well!

Arup has proposed to divide the land using the line y = mx, giving Matt the triangle formed using the leg of the original triangle on the x-axis, while Arup gets the other triangle formed using the leg of the original triangle on the y-axis:



Matt would like your help in evaluating Arup's offer. Write a program that calculates the area of Arup's land and Matt's land, under Arup's proposal.

Input

The input contains 3 positive real numbers, ***a***, ***b***, (1 ≤ ***a***, ***b*** ≤ 1000)***,*** and ***m*** (0.1 ≤ ***m*** ≤ 10), representing the lengths of the legs of the right triangle of land given to Arup, and the slope of the dividing line proposed by Arup. Specifically, the triangle has the vertices (0 0), (***a***,0) and (0,***b***), and the dividing line is y = ***m***x.

Output

On a single line, output the area of Matt's land, rounded to 3 decimal places, followed by a space, followed by the area of Arup's land, rounded to 3 decimal places.

Samples

Input	Output
`20 10 1`	`66.667 33.333`
`3.4 10.2 .4`	`2.040 15.300`