Objects and Classes – Lab

This document defines the lab for the "C++ OOP" course @ Software University. Please submit your solutions (source code) to all below-described problems in Judge.

Write C++ code for solving the tasks on the following pages.

Please try to solve the problems using **classes and objects**.

Keep in mind the type of submission if **file upload**.

1. Letters

You are given a text in English. Let's define a word as any sequence of alphabetical characters. Each of those characters we will call a letter, but we will consider the uppercase and lowercase variant of a character in a word as the **same** letter.

Write a program that reads the text (a single line on the console) and then reads lines, each containing a single letter, until a line containing a **'.'** (dot) is entered. For each of those lines, print all words that contain the letter, ordered alphabetically (capitals letters before lowercase letters), without duplicates – if no words contain that letter, print **"---"** (three dashes)

Examples

Input (NOTE: the *italic* text is on a single line)
*You are given a text in English. Let's define a word as any sequence of alphabetical characters. Each of those characters we'll call a letter, but we will consider the uppercase and lowercase variant of a character in a word as the **same** letter.* a Y h .

Output
Each a alphabetical and any are as call character characters lowercase same uppercase variant You any Each English alphabetical character characters the those

2. Rust

You are given a 10x10 matrix representing a metal square, which has begun to rust. There are 3 types of symbols in the matrix – a **.** (dot) means a healthy part of the metal, a **#** indicates a rust-resistant part, and a **!** indicates a part that has begun to rust.

There may be **0, 1 or more** parts that have begun to rust.

The rust spreads from a rusty cell to healthy cells by "infecting" adjacent cells directly above, to the right, below, and to the left of itself **(no diagonals)**, at the same time. The rust cannot infect cells that are indicated as **rust resistant**. Let's define the time it takes for all cells adjacent to a rusty cell to get infected as 1 unit.

After reading the matrix, read a single integer – the elapsed time in units (as defined above) – and print a matrix representing how the metal square will look after the rust has been acting on it for that amount of time.

Examples

Input	Output	Input	Output	Input	Output
....!.... 4	.!!!!!!!.. !!!!!!!!!. .!!!!!!!.. ..!!!!!... ...!!!.... ....!.....	....!.... ...###.... 5	!!!!!!!!!. !!!!!!!!!! !!!###!!!. .!!!.!!!.. ..!...!... 	!........! .... !........! 5	!!!!!!!!!! !!!!!!!!!! !!!!..!!!! !!!....!!! !!......!! !!......!! !!!....!!! !!!!..!!!! !!!!!!!!!! !!!!!!!!!! 5

3. Matching Locations

Write a program that reads **names** of places and their geographical **coordinates** in the format **`name,latitude,longitude`** (where latitude and longitude are floating-point numbers). No two locations will have the same **name**. Some locations may have the same **coordinates**.

After all locations are entered, a single line containing the **`'.'`** (dot) the character will be entered.

After that, queries will be entered – the queries will either contain a **name** of a location, or **`latitude`** and **`longitude`** coordinates (entered as two floating point numbers separated by a single space). Print all locations that match the query in the same format that they were entered.

After all queries are entered, a single line containing the **`'.'`** (dot) the character will be entered.

Examples

Input	Output
Sofia,42.70,23.33 New York,40.6976701,-74.2598732 SoftUni,42.70,23.33 . Sofia	Sofia,42.70,23.33 New York,40.6976701,-74.2598732 Sofia,42.70,23.33 SoftUni,42.70,23.33

40.6976701 -74.2598732 42.70 23.33 .	