Exercises: Arrays

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

1. Compare Arrays

Write a program that reads two arrays of Integers and compares them element by element. For better code reusability, you could do the comparison in a

`bool areEqual(int arr1[], int length1, int arr2[], int lenght2)` function, which returns TRUE if they are equal and FALSE if not. Each array will be defined by two lines on the console – the first containing a number representing the length of the array, and the second containing the numbers in the array.

Print **equal** if the arrays match, and **not equal** if the arrays don't match.

Examples

Input	Output
3 1 2 3 3 1 2 3	equal
3 1 2 3 2 2 1	Not equal

2. Longest Sequence

Write a program that finds the longest sequence of equal elements in an integer array and then prints that sequence on the console (integers separated by space on a single line). If there is more than one such sequence, print the last one. The input array will be entered on two lines – the first line will contain an integer representing the number of elements, the second will contain the elements separated by spaces.

Examples

Input	Output
7 13 10 10 1 4 2 10	10 10
5 13 42 19 21 103	103

3. Above Average

Write a program that reads an array of integer numbers from the console and prints all numbers which are larger than or equal to the mathematical average of the numbers in the array. The output should be printed on a single line, separating the output number by spaces. The output numbers should be in the same order as they were in the input.

The output array will be entered on two lines- the first line will contain an integer representing the number of elements, the second will contain the elements separated by spaces.

Examples

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

4. Most Frequent Number

Write a program that finds the most frequent number in a given sequence of numbers. Numbers will be in the range [0, 9]. In case of multiple numbers with the same maximal frequent, print all of them, ordered, from smallest to largest, separated by spaces.

Examples

Input	Output	Comments
13 4 1 1 4 2 3 4 4 1 2 4 9 3	4	The number 4 is the most frequent
8 2 2 2 2 1 2 2 2	2	The number 2 is the most frequent

5. Cartesian Product

Write a program that reads an array from the console and prints the product of each of its elements with all elements. E.g. for the array {1, 7, 3}, the result would be

{1*1, 1*7, 1*3, 7*1, 7*7, 7*3, 3*1, 3*7, 3*3},

which gives us the array

{1, 7, 3, 7, 49, 21, 3, 21, 9},

so for the input

1 7 3,

the program should print

1 7 3 7 49 21 3 21 9.

Examples

Input	Output
3 1 7 3	1 7 3 7 49 21 3 21 9
2 -1 4	1 -4 -4 16

6. Closest Numbers

Write a program that finds the two closest (by value) integer numbers in an array and prints the absolute difference between them.

Examples

Input	Output	Comments
5 1 105 10 100 3	2	The closest numbers are 1 and 3, abs(1,3) = 2
9 1 2 3 4 5 6 7 8 9	1	All numbers are exactly 1 unit apart

7. Noise

Write a program that reads a positive integer number and returns its square root (print the result as **`cout`** prints **`double`** numbers)

The number will be entered with "noise" in it, i.e. there will be symbols that are not digits. These symbols should be ignored. The last symbol of the input of the number will always be **`.`** (dot) and there will be no other **`.`** (dot) in the number.

Examples

Input	Output
25.	5
,,-2!!as**dsa5*-.	5
-9abc.	3

8. Noise Test Generator* **(not included in the homework)**

Write a program that generates test input and expected output data for Problem 7 – Noise. The program should generate a random input value on each execution, print that input value on the console on a single line, then print another line with the expected output value for that input. Hint: lookup **`srand()`** and **`rand()`**.

Examples

Output
25. 5
,,-2!!as**dsa5*-. 5
-9abc. 3