

Homework: Sorting and Searching Algorithms

This document defines the **homework assignments** for the ["Algorithms" course @ Software University](#). Please submit a single **zip / rar / 7z** archive holding the solutions (source code) of all below described problems. You can check your solutions here: <https://judge.softuni.bg/Contests/690/Sorting-and-Searching>

1. Sorting

Implement any or many of the following sorting algorithms:

- Insertions Sort
- Bubble Sort
- Shell Sort
- Merge Sort
- Quicksort
- Bucket Sort

Read a sequence of numbers from the console, sort them and print them back.

Examples

Input	Output
5 4 3 2 1	1 2 3 4 5
1 4 2 -1 0	-1 0 1 2 4

Here is a breakdown of judge tests:

Test N	N of Elements	Comments
001	1k	Random
002		Sorted
003		Reversed
004		Duplicate
005	10k	Random
006		Sorted
007		Reversed
008		Duplicate
009	100k	Random
010		Sorted
011		Reversed
012		Duplicate

2. Searching

Implement any or many of the following sorting algorithms:

- Linear Search

- Binary Search
- Fibonacci Search

Read a sequence of numbers on the first line and a single number on the second from the console. Find the index of the number in the given array. Return -1 if the element is not present in the array.

Examples

Input	Output
1 2 3 4 5 1	0
1 2 3 4 5 6	-1

3. Inversion Count

Assume an inversion count is how far (or close) the array is from being sorted. If array is already sorted then inversion count is 0. If array is sorted in reverse order then inversion count is at its maximum.

Two elements $a[i]$ and $a[j]$ form an inversion if $a[i] > a[j]$ and $i < j$.

Find and **print the count of all inversions** in a given input array.

Examples

Input	Output	Inversions
2 4 1 3 5	3	2 1 4 1 4 3
5 4 3 2 1	10	5 4 5 3 5 2 5 1 4 3 4 2 4 1 3 2 3 1

Hints

Use a modified version of merge sort.

Useful read: <http://www.geeksforgeeks.org/counting-inversions/>

4. * Words

You can test your solution to the problem in the Judge system [here](#).

You are given a string containing Latin letters. Write a program that finds **the number of all words with no two consecutive equal characters that can be generated by reordering the given letters**. The generated words should contain all given letters. If the given word meets the requirements it should also be considered in the count.

Input

- The input data should be read from the console.
- On the only input line, there will be a single word containing all the letters that you should use for generating the words.
- The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

- The output data should be printed on the console.
- On the only output line write the number of words found.

Constraints

- The number of the given letters will be between 1 and 10, inclusive.
- All given letters will be small Latin letters ('a' – 'z')
- Allowed working time for your program: 0.35 seconds. Allowed memory: 32 MB.

Examples

Input	Sample Output	Comments
xy	2	Two possible words: "xy" and "yx"
xxxy	0	It is impossible to construct a word with these letters.
aahhhaa	1	The only possible word is "ahahaha".
nopqrstuvw	3628800	There are 3628800 possible words.

5. * Needles

You can test your solution to the problem in the Judge system [here](#).

This problem is about finding the proper place of numbers in an array. From the console, you'll read a sequence of non-decreasing integers with randomly distributed "holes" among them (represented by zeros).

Then you'll be given the needles – numbers which should be inserted into the sequence, so that it remains non-decreasing (discounting the "holes"). For each needle, find the left-most index where it can be inserted.

Input

- The input should be read from the console.
- On the first line you'll be given the numbers C and N separated by a space.
- On the second line you'll be given C non-negative integers forming a non-decreasing sequence (disregarding the zeros).
- On the third line you'll be given N positive integers, the needles.
- The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

- The output should be printed on the console. It should consist of a single line.
- On the only output line print N numbers separated by a space. Each number represents the left-most index at which the respective needle can be inserted.

Constraints

- All input numbers will be 32-bit signed integers.
- N will be in the range [1 ... 1000].
- C will be in the range [1 ... 50000].
- Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

Examples

Input
23 9 3 5 11 0 0 0 12 12 0 0 0 12 12 70 71 0 90 123 140 150 166 190 0 5 13 90 1 70 75 7 188 12
Output
1 13 15 0 13 15 2 21 3
Comments
5 goes to index 1 - between 3 and 5 13 goes to index 13 - 12 and 70 90 goes to index 15 - between 71 and 0 1 goes to index 0 - before 3 Etc.
Input
11 4 2 0 0 0 0 0 0 0 0 0 3 4 3 2 1
Output
11 1 0 0