

Эта тренировка предназначена для изучения базовых конструкций языков С и С++. Поэтому её условия более строгие, чем в пробной тренировке.

Во-первых, задачи получится послать на проверку только на языках С и С++. Во-вторых, не следует пользоваться библиотеками языка, кроме библиотеки ввода-вывода `stdio.h` (или `cstdio`). Решения, подключающие другие библиотеки, не будут считаться правильными при итоговом подсчёте баллов.

## Задача А. Треугольник

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан прямоугольник  $p \times q$  метров. Найдите треугольник максимальной возможной площади, содержащийся внутри него.

### Формат входных данных

В первой строке ввода заданы два целых числа  $p$  и  $q$  через пробел — длины сторон прямоугольника ( $1 \leq p, q \leq 100$ ).

### Формат выходных данных

В первой строке выведите одно число — максимальную площадь треугольника. В следующих трёх строках выведите по два числа  $x_i$  и  $y_i$  в каждой через пробел — координаты вершин треугольника, имеющего максимальную площадь. Порядок вывода вершин значения не имеет. Координаты должны удовлетворять соотношениям  $0 \leq x_i \leq p$ ,  $0 \leq y_i \leq q$ . Все числа следует выводить не менее чем с шестью точными знаками после запятой. Если треугольников максимальной площади несколько, можно выводить любой из них.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 1	0.5 0.633333 1.000000 0.000000 0.000000 1.000000 0.000000

## Задача В. Два числа

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Даны два целых числа  $A$  и  $B$  ( $1 \leq A, B \leq 100$ ). Найдите два таких целых числа  $X$  и  $Y$ , что выполнено равенство  $AX + BY = 1$ .

### Формат входных данных

В первой строке записаны два числа  $A$  и  $B$ , разделённые пробелом.

### Формат выходных данных

Выведите два числа  $X$  и  $Y$ , разделённые пробелом. Требуется, чтобы выполнялись неравенства  $|X| \leq 10\,000$ ,  $|Y| \leq 10\,000$ . Если правильных ответов несколько, разрешается вывести любой из них. Если же таких чисел не существует, выведите вместо них два нуля.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3	2 -1
4 6	0 0
100 51	-5075 9951

## Задача С. Проверка на простоту

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Натуральное число называется *простым*, если оно делится нацело только на себя и на 1. Единица простым числом не считается.

Дано число  $p$ . Определите, простое ли оно.

### Формат входных данных

В первой строке задано натуральное число  $p$  ( $1 \leq p \leq 10^9$ ).

### Формат выходных данных

В первой строке выведите «YES», если число простое, и «NO» в противном случае.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	YES
6	NO

## Задача D. Сортировка

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В этой задаче нужно отсортировать заданные числа.

### Формат входных данных

В первой строке задано целое число  $n$  ( $1 \leq n \leq 300\,000$ ). Во второй строке заданы через пробел  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 1000$ ).

### Формат выходных данных

В первой строке выведите через пробел  $n$  чисел — числа  $a_i$  в неубывающем порядке.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 3	1 2 3
4 3 2 2 1	1 2 2 3
5 10 100 10 1000 10	10 10 10 100 1000

## Задача Е. Подстрока

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

*Подстрокой* строки  $S = s_1s_2 \dots s_n$  называется непрерывная подпоследовательность символов этой строки  $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$ .

Даны две строки. Сколько раз вторая из них встречается в первой как подстрока?

### Формат входных данных

В первой строке ввода задана строка длины от 1 до 100 символов, включительно. Во второй строке задана строка длины от 1 до 10 символов, включительно. Строки состоят из маленьких букв английского алфавита.

### Формат выходных данных

В первой строке выведите одно число — количество вхождений второй строки в первую как подстроки.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
aab ab	1
abababc aba	2

## Задача F. Различные подстроки

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

*Подстрокой* строки  $S = s_1s_2 \dots s_n$  называется непрерывная подпоследовательность символов этой строки  $s_i s_{i+1} s_{i+2} \dots s_{j-1} s_j$ .

Дана строка. Сколько различных подстрок, не считая пустой, она содержит?

### Формат входных данных

В первой строке ввода задана строка длины от 1 до 100 символов, включительно. Строка состоит из маленьких букв английского алфавита.

### Формат выходных данных

В первой строке выведите одно число — количество различных подстрок данной строки, не считая пустой.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
aab	5
dabux	15

## Задача G. Дерево

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф. Проверьте, является ли он деревом.

### Формат входных данных

В первой строке заданы через пробел два целых числа  $n$  и  $m$  — количество вершин и рёбер в графе, соответственно ( $1 \leq n \leq 100$ ). В следующих  $m$  строках заданы рёбра;  $i$ -я из этих строк содержит два целых числа  $u_i$  и  $v_i$  через пробел — номера концов  $i$ -го ребра ( $1 \leq u_i, v_i \leq n$ ). Граф не содержит петель и кратных рёбер.

### Формат выходных данных

В первой строке выведите «YES», если граф является деревом, и «NO» в противном случае.

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 2 1 2 1 3	YES
3 3 1 2 2 3 3 1	NO

## Задача Н. Ханойские башни

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Ханойские башни – популярная головоломка. Она состоит из трёх стержней и  $n$  дисков различного диаметра. В центре каждого диска находится отверстие для того, чтобы нанизывать диск на стержень. Изначально все диски нанизаны на первый стержень, причём сверху расположен самый маленький диск, под ним диск побольше и так далее; снизу лежит самый большой диск. За одно перекаладывание разрешается снять один диск сверху любого стержня и нанизать его сверху на другой стержень. При этом нельзя нанизывать диск на стержень, на котором верхний диск имеет меньший диаметр; на пустой стержень можно нанизывать любой диск. Цель – перекаладывать диски таким образом, чтобы перенести их все на третий стержень.

Напишите программу, которая по заданному количеству дисков выдаст последовательность перекаладываний, позволяющую решить головоломку. Количество перекаладываний должно быть минимально возможным.

### Формат входных данных

В первой строке записано целое число  $n$  – количество дисков ( $1 \leq n \leq 21$ ).

### Формат выходных данных

В первой строке выведите  $m$  – минимальное число перекаладываний. В следующих  $m$  строках выведите описания операций перекаладывания по одному на строке. Описание перекаладывания должно состоять из двух целых чисел, разделённых пробелом – номеров стержня, с которого снимается диск, и стержня, на который он нанизывается. Если существует несколько способов решить головоломку за минимальное число перекаладываний, можно выводить любой из них.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	3 1 2 1 3 2 3

## Задача I. Ханойские башни 2

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ханойские башни – популярная головоломка. Она состоит из трёх стержней и  $n$  дисков различного диаметра. В центре каждого диска находится отверстие для того, чтобы нанизывать диск на стержень. Изначально все диски нанизаны на первый стержень, причём сверху расположен самый маленький диск, под ним диск побольше и так далее; снизу лежит самый большой диск. За одно перекаладывание разрешается снять один диск сверху любого стержня и нанизать его сверху на другой стержень. При этом нельзя нанизывать диск на стержень, на котором верхний диск имеет меньший диаметр; на пустой стержень можно нанизывать любой диск. Цель – перекаладывать диски таким образом, чтобы перенести их все на третий стержень.

Напишите программу, которая решает головоломку в общем виде: по заданному количеству дисков и данному начальному положению находит последовательность перекаладываний, позволяющую перевести их в данное конечное положение. Количество перекаладываний должно быть минимально возможным.

### Формат входных данных

В первой строке записано целое число  $n$  – количество дисков ( $1 \leq n \leq 15$ ). Во второй строке записано  $n$  целых чисел через пробел – номера стержней, на которых изначально лежат диски. В третьей строке также записано  $n$  целых чисел через пробел – номера стержней, на которых должны оказаться диски. В этих строках диски перечислены от маленьких к большим. Если несколько дисков в начальном или конечном положении лежат на одном стержне, это означает, что внизу лежит самый большой из них, на нём – самый большой из оставшихся, и так далее; наверху лежит самый маленький диск.

### Формат выходных данных

В первой строке выведите  $m$  – минимальное число перекаладываний. В следующих  $m$  строках выведите описания операций перекаладывания по одному на строке. Описание перекаладывания должно состоять из двух целых чисел, разделённых пробелом – номеров стержня, с которого снимается диск, и стержня, на который он нанизывается. Если существует несколько способов решить головоломку за минимальное число перекаладываний, можно выводить любой из них. Если решения не существует, выведите в первой строке число  $-1$ .

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1 1 3 3	3 1 2 1 3 2 3
3 1 2 3 2 3 1	5 1 2 3 1 2 1 2 3 1 2

## Задача J. Перекладывание камней

Имя входного файла: *стандартный ввод*  
Имя выходного файла: *стандартный вывод*  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

На столе лежит  $n$  кучек камней —  $k_1$  камней в первой кучке,  $k_2$  камней во второй,  $\dots$ ,  $k_n$  в последней. Каждую секунду выбирается две кучки: кучка, содержащая максимальное количество камней  $M$ , и кучка, содержащая минимальное количество камней  $m$ . Если  $M - m > 1$ , то из кучки с  $M$  камнями берут  $\lfloor \frac{M}{2} \rfloor$  камней и перекладывают в кучку, содержащую  $m$  камней. В противном случае процесс перекладывания завершается.

Через сколько секунд после начала перекладывания на столе получится такая конфигурация, в которой максимальное и минимальное количества камней в кучках отличаются не более чем на единицу, и процесс завершится?

### Формат входных данных

В первой строке задано целое число  $n$  — количество кучек камней на столе ( $1 \leq n \leq 60$ ). Вторая строка содержит  $n$  целых чисел  $k_1, k_2, \dots, k_n$  через пробел — количества камней в кучках ( $1 \leq k_i \leq 60$ ). Общее количество камней на столе не превосходит 60.

### Формат выходных данных

В первой строке выведите одно число — через сколько секунд после начала перекладывания количества камней в кучках, находящихся на столе, будут отличаться не более чем на единицу. Если процесс перекладывания никогда не завершится, выведите  $-1$ .

### Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 3	1
3 1 4 1	2
4 4 4 4 4	0
2 4 2	-1