

## Содержание

|   |          |
|---|----------|
| <b>Easy</b>                                       | <b>2</b> |
| Задача А. Заезд в ЛКШ [1 sec, 256 mb]             | 2        |
| Задача В. Бублики [2 sec, 256 mb]                 | 3        |
| <b>Medium</b>                                     | <b>4</b> |
| Задача С. Различные слагаемые [2 sec, 256 mb]     | 4        |
| Задача D. Белоснежка и $n$ гномов [1 sec, 256 mb] | 5        |
| Задача Е. Скалярное произведение [2 sec, 256 mb]  | 6        |
| Задача F. Раскраска доски [2 sec, 256 mb]         | 7        |
| <b>Hard</b>                                       | <b>8</b> |
| Задача G. Красивый граф [2 sec, 256 mb]           | 8        |

---

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

**Обратите внимание**, что ввод-вывод во всех задачах стандартный.

## Easy

### Задача А. Заезд в ЛКШ [1 sec, 256 mb]

Знаете ли вы, как непросто организовать заезд в ЛКШ? Например, в 2013 году нужно было заказать автобусы для целых  $n$  ЛКШат, мечтающих добраться в “Берендеевы поляны” из Москвы. Один из директоров ЛКШ сообщил другому директору, что можно заказать некоторые из  $m$  автобусов. Он узнал вместимость каждого автобуса и сразу понял, какое минимальное количество автобусов ему нужно заказать, чтобы привезти в лагерь всех ЛКШат. А сможете ли вы так же быстро решить эту задачу?

#### Формат входных данных

В первой строке через пробел записаны целые числа  $n$  и  $m$  ( $1 \leq n \leq 10^6$ ;  $1 \leq m \leq 1000$ ). В следующей строке через пробел записаны  $m$  целых чисел в пределах от 1 до 1000 — вместимости автобусов.

#### Формат выходных данных

В первой строке выведите число  $k$  — минимальное количество автобусов, которое придётся заказать директору. В следующей строке выведите через пробел  $k$  целых чисел — номера автобусов, которые нужно заказать. Автобусы пронумерованы от 1 до  $m$  в том порядке, в которых они перечислены во входных данных. Если возможных решений несколько, выведите любое. Если решения нет, в единственной строке выведите “-1”.

#### Примеры

| stdin                        | stdout     |
|------------------------------|------------|
| 345 5<br>100 130 190 140 150 | 3<br>1 3 4 |
| 345 3<br>100 100 100         | -1         |

### Задача В. Бублики [2 sec, 256 mb]

В сказочной стране пекарей деньги традиционно называются «бубликами». В этой стране в ходу  $n$  различных монет номиналом в  $v_1, v_2, \dots, v_n$  бубликов. Номиналы таковы, что для любых двух монет достоинство одной из них делится нацело на достоинство другой.

Вас попросили написать программу для автомата, выдающего суммы бубликов монетами. Для удобства пользователей автомат должен выдавать любую сумму при помощи минимально возможного для этой суммы количества монет.

Считая, что количество монет каждого достоинства в автомате не ограничено, выясните, как ему выдать  $w$  бубликов минимальным количеством монет.

#### Формат входных данных

В первой строке входного файла заданы два целых числа  $n$  и  $w$  через пробел — количество различных достоинств монет и сумма, которую необходимо набрать, соответственно ( $1 \leq n \leq 30, 1 \leq w \leq 10^9$ ). Во второй строке заданы  $n$  различных целых чисел  $v_1, v_2, \dots, v_n$  через пробел — достоинства монет ( $1 \leq v_i \leq 10^9$ ). Известно, что для любых двух различных достоинств одно из них делится на другое. Одно из заданных достоинств равно единице.

#### Формат выходных данных

В первой строке выходного файла выведите одно число  $k$  — минимальное количество монет, при помощи которого можно набрать необходимую сумму. Во второй строке перечислите достоинства этих  $k$  монет в любом порядке; сумма этих достоинств должна быть равна  $w$ . Если оптимальных ответов несколько, можно вывести любой из них. Тесты в этой задаче таковы, что в оптимальном ответе число  $k$  не превысит 10 000.

#### Примеры

| stdin             | stdout              |
|-------------------|---------------------|
| 4 90<br>1 2 10 50 | 5<br>10 10 10 10 50 |
| 4 3<br>1 2 10 50  | 2<br>2 1            |

## Medium

### Задача С. Различные слагаемые [2 сек, 256 mb]

В этой задаче требуется найти разбиение натурального числа  $n$  на как можно большее количество различных натуральных слагаемых.

#### Формат входных данных

В первой строке входного файла записано натуральное число  $n$  ( $1 \leq n \leq 1\,000\,000\,000$ ).

#### Формат выходных данных

В первой строке выходного файла выведите  $k$  — количество слагаемых. Во второй строке выведите  $k$  чисел через пробел — сами слагаемые. Если разбиений с максимальным количеством слагаемых несколько, можно вывести любое из них. Слагаемые можно выводить в любом порядке.

#### Примеры

| stdin | stdout     |
|-------|------------|
| 4     | 2<br>1 3   |
| 6     | 3<br>1 2 3 |

### Задача D. Белоснежка и $n$ гномов [1 sec, 256 mb]

«Ну не гномы, а наказание какое-то!», — подумала Белоснежка, в очередной раз пытаясь уложить гномов спать. Одного уложишь — другой уже проснулся! И так всю ночь.

У Белоснежки  $n$  гномов, и все они очень разные. Она знает, что для того, чтобы уложить спать  $i$ -го гнома нужно  $a_i$  минут, и после этого он будет спать ровно  $b_i$  минут. Помогите Белоснежке узнать, может ли она получить хотя бы минутку отдыха, когда все гномы будут спать, и если да, то в каком порядке для этого нужно укладывать гномов спать.

Например, пусть есть всего два гнома,  $a_1 = 1$ ,  $b_1 = 10$ ,  $a_2 = 10$ ,  $b_2 = 20$ . Если Белоснежка сначала начнет укладывать первого гнома, то потом ей потребуется целых 10 минут, чтобы уложить второго, а за это время проснется первый. Если же она начнет со второго гнома, то затем она успеет уложить первого и получит целых 9 минут отдыха.

#### Формат входных данных

Первая строка входного файла содержит число  $n$  ( $1 \leq n \leq 10^5$ ), вторая строка содержит числа  $a_1, a_2, \dots, a_n$ , третья — числа  $b_1, b_2, \dots, b_n$  ( $1 \leq a_i, b_i \leq 10^9$ ).

#### Формат выходных данных

Выведите в выходной файл  $n$  чисел — порядок, в котором нужно укладывать гномов спать. Если Белоснежке отдохнуть не удастся, выведите число  $-1$ .

#### Пример

| stdin               | stdout |
|---------------------|--------|
| 2<br>1 10<br>10 20  | 2 1    |
| 2<br>10 10<br>10 10 | -1     |
| 3<br>1 4 1<br>5 3 4 | 2 1 3  |

### Задача Е. Скалярное произведение [2 sec, 256 mb]

Даны два вектора:  $v_1 = (x_1, x_2, \dots, x_n)$  и  $v_2 = (y_1, y_2, \dots, y_n)$ . Скалярным произведением этих векторов называется значение, вычисляемое по формуле:  $x_1y_1 + x_2y_2 + \dots + x_ny_n$ .

Разрешено переставлять координаты каждого из векторов любым образом. Выберите такие их перестановки, чтобы скалярное произведение двух полученных векторов было минимальным и выведите его значение.

$1 \leq n \leq 800$ .  $-100\,000 \leq x_i, y_i \leq 100\,000$ .

#### Формат входных данных

Первая строка входного файла содержит единственное целое число  $t$  — количество наборов тестовых данных. Далее следуют сами наборы, по три строки в каждом. Первая строка тестового набора содержит единственное целое число  $n$ . Две следующие строки содержат по  $n$  целых чисел, задающих координаты соответствующего вектора, каждая.

#### Формат выходных данных

Для каждого набора выведите строку с номером этого набора и ответом на задачу — значением минимального скалярного произведения. Следуйте формату, указанному в примере.

#### Пример

| stdin     | stdout       |
|-----------|--------------|
| 2         | Case #1: -25 |
| 3         | Case #2: 6   |
| 1 3 -5    |              |
| -2 4 1    |              |
| 5         |              |
| 1 2 3 4 5 |              |
| 1 0 1 0 1 |              |

### Задача F. Раскраска доски [2 sec, 256 mb]

Дана прямоугольная доска  $w \times h$ , состоящая из квадратов  $1 \times 1$ . Некоторые из квадратов отсутствуют. Нужно покрасить оставшиеся квадраты в чёрный и белый так, чтобы число чёрных было как можно больше, и чтобы никакие два одноцветных квадрата не имели общей стороны.

#### Формат входных данных

В первой строке ввода записаны числа  $w$  и  $h$  — размеры доски ( $1 \leq w, h \leq 50$ ). В следующих  $h$  строках записано по  $w$  символов: «.» означает, что соответствующий квадрат отсутствует, «#» — обратное.

#### Формат выходных данных

Выведите доску, раскрашенную указанным образом. Среди всех оптимальных решений выведите лексикографически минимальное. Чёрный квадрат обозначается символом «b», белый — «w».

#### Примеры

| stdin   | stdout   |
|---|--|
| 3 3<br>.#.<br>###<br>.#.  | .b.<br>bwb<br>.b.  |
| 6 6<br>#.#.#.<br>.#.###<br>#.#.#.<br>.#.###<br>#.#.#.<br>#.#.#. | b.b.b.<br>.b.b.b<br>b.b.b.<br>.b.b.b<br>b.b.b.<br>.b.b.b |
| 6 1<br>#####  | bwbwbw   |
| 3 4<br>.#.<br>.#.<br>###<br>.#.                                 | .w.<br>.b.<br>bwb<br>.b.                                 |

## Hard

### Задача G. Красивый граф [2 sec, 256 mb]

Дан неориентированный граф из  $N$  вершин. Каждая вершина либо красная, либо чёрная. Назовем граф красивым, если ни из какой красной вершины не достижима никакая другая красная вершина. Требуется добавить в граф максимальное количество рёбер, чтобы он остался красивым.

#### Формат входных данных

В первой строке входного файла будет записано одно целое число  $1 \leq N \leq 50$ . Следующие  $N$  строк будут содержать по  $N$  символов каждая,  $i$ -ый символ  $j$ -ой строки будет равен 1, если существует ребро между  $i$ -й и  $j$ -й вершиной, и 0 иначе. Следующая строка будет содержать одно целое число  $1 \leq M \leq N$  — количество красных вершин. Затем будут следовать  $M$  целых различных чисел — номера красных вершин. Вершины во входном файле нумеруются с нуля. Гарантируется, что исходный граф красивый.

#### Формат выходных данных

Выходной файл должен содержать единственное число — максимальное количество рёбер, которое можно добавить в граф так, чтобы он остался при этом красивым.



**Примеры**

| stdin  | stdout |
|--|--------|
| 3<br>000<br>000<br>000<br>1<br>0                                 | 3      |
| 3<br>000<br>000<br>000<br>2<br>0 1                               | 1      |
| 2<br>01<br>10<br>1<br>0  | 0      |
| 5<br>00000<br>00000<br>00000<br>00000<br>00000<br>5<br>0 1 2 3 4 | 0      |
| 5<br>01000<br>10100<br>01010<br>00100<br>00000<br>2<br>2 4       | 3      |