

## Задача А. Выбор заявок

Имя входного файла: `request.in`  
Имя выходного файла: `request.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Вы прекрасно знаете, что в ЛКШ.Зима 2016 лекции читают лучшие преподаватели мира. К сожалению, лекционных аудиторий у нас не так уж и много, поэтому каждый преподаватель составил список лекций, которые он хочет прочитать ЛКШатам. Чтобы ЛКШата, утром идя на завтрак, увидели расписание лекций, необходимо его составить прямо сейчас. И без вас нам здесь не справиться.

У нас есть список заявок от преподавателей на лекции для одной из аудиторий. Каждая заявка представлена в виде временного интервала  $[s_i, f_i)$  — время начала и конца лекции. Лекция считается открытым интервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Необходимо выбрать из этих заявок такое подмножество, чтобы суммарно выполнить максимальное количество заявок. Учтите, что одновременно в лекционной аудитории, конечно же, может читаться лишь одна лекция.

### Формат входных данных

В первой строке вводится натуральное число  $N$ , не более 1000 — общее количество заявок на лекции. Затем вводится  $N$  строк с описаниями заявок — по два числа в каждом  $s_i$  и  $f_i$ . Гарантируется, что  $s_i < f_i$ . Время начала и окончания лекции — натуральные числа, не превышают 1440 (в минутах с начала суток).

### Формат выходных данных

Выведите одно число — максимальное количество заявок, которые можно выполнить.

### Примеры

<code>request.in</code>	<code>request.out</code>
1 5 10	1
3 1 5 2 3 3 4	2

## Задача В. Последовательность

Имя входного файла: `sequence.in`  
Имя выходного файла: `sequence.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дана последовательность натуральных чисел  $a_1, a_2, \dots, a_n$ , и известно, что  $a_i \leq i$  для любого  $1 \leq i \leq n$ . Требуется определить, можно ли разбить элементы последовательности на две части таким образом, что сумма элементов в каждой из частей будет равна половине суммы всех элементов последовательности.

### Формат входных данных

В первой строке входного файла находится одно целое число  $n$  ( $1 \leq n \leq 40\,000$ ). Во второй строке находится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq i$ ).

### Формат выходных данных

В первую строку выходного файла выведите количество элементов последовательности в любой из получившихся двух частей, а во вторую строку через пробел номера этих элементов. Если построить такое разбиение невозможно, выведите -1.

### Примеры

<code>sequence.in</code>	<code>sequence.out</code>
3 1 2 3	1 3

## Задача С. Индикатор

Имя входного файла: `indic.in`  
Имя выходного файла: `indic.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Недавно Вася приобрел настольный калькулятор с жидкокристаллическим индикатором. Этот индикатор отображает  $n$  цифр с помощью  $n$  одинаковых элементов.



Отметим, что каждый элемент содержит семь полосок, каждая из которых может быть либо белой, либо черной. В частности, при отображении цифры «1» черными являются две полоски.

Вася — очень любознательный мальчик, поэтому он хочет узнать, какое максимальное и минимальное  $n$ -значное число может быть отображены на индикаторе его нового калькулятора так, чтобы черными были ровно  $k$  полосок.

Напишите программу, которая найдет ответ на Васин вопрос. Учитывайте при этом, что числа не могут содержать ведущие нули.

### Формат входных данных

Первая строка входного файла содержит два целых числа:  $n$  и  $k$  ( $1 \leq n \leq 100$ ,  $1 \leq k \leq 700$ ).

### Формат выходных данных

В первой строке выходного файла выведите минимальное число, во второй строке выходного файла выведите максимальное число. Если указанным образом не может быть представлено ни одно число, выходной файл должен содержать одну строку `NO SOLUTION`.

### Примеры

<code>indic.in</code>	<code>indic.out</code>
5 15	10117 97111
10 1	NO SOLUTION

## Задача D. Коммерческий калькулятор

Имя входного файла: `calc.in`  
Имя выходного файла: `calc.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Фирма OISAC выпустила новую версию калькулятора. Этот калькулятор берет с пользователя деньги за совершаемые арифметические операции. Стоимость каждой операции в долларах равна 5% от числа, которое является результатом операции.

На этом калькуляторе требуется вычислить сумму  $N$  натуральных чисел (числа известны). Нетрудно заметить, что от того, в каком порядке мы будем складывать эти числа, иногда зависит, в какую сумму денег нам обойдется вычисление суммы чисел (тем самым, оказывается нарушен классический принцип "от перестановки мест слагаемых сумма не меняется":-) ).

Например, пусть нам нужно сложить числа 10, 11, 12 и 13. Тогда если мы сначала сложим 10 и 11 (это обойдется нам в \$1.05), потом результат - с 12 (\$1.65), и затем - с 13 (\$2.3), то всего мы заплатим \$ 5, если же сначала отдельно сложить 10 и 11 (\$1.05), потом - 12 и 13 (\$1.25) и, наконец, сложить между собой два полученных числа (\$2.3), то в итоге мы заплатим лишь \$4.6.

Напишите программу, которая будет определять, за какую минимальную сумму денег можно найти сумму данных  $N$  чисел.

### Формат входных данных

Во входном файле записано число  $N$  ( $2 \leq N \leq 1000$ ). Далее идет  $N$  натуральных чисел, которые нужно сложить, каждое из них не превышает 10000.

### Формат выходных данных

В выходной файл выведите, сколько денег нам потребуется на нахождение суммы этих  $N$  чисел. Результат должен быть выведен с двумя знаками после десятичной точки.

### Примеры

<code>calc.in</code>	<code>calc.out</code>
4 10 11 12 13	4.60
2 1 1	0.10