

## Задача 1. Переливание духов

Входной файл	flow.in
Выходной файл	flow.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_1.pas

...но нужен ли на самом деле кому-нибудь  $6\frac{1}{7}$ -пунктовый шрифт, который на три четверти — шрифт Baskerville, и на четверть — Helvetica?

Дональд Э. Кнут. Идея Мета-фонта<sup>1</sup>

Маленькая девочка Оля добралась до полки, на которой стоят флаконы с мамиными любимыми духами, и начала заниматься своим любимым занятием — переливанием жидкостей из одного флакона в другой. Когда наконец мама застала её за этим занятием, прекрасная коллекция духов была уже безнадёжно утрачена.

К счастью, Оля — аккуратная девочка, поэтому все свои действия она записывала на бумажку. Помогите ей успокоить маму: определите, каков состав духов в первом (мамино любимом) флаконе, чтобы мама смогла придумать этой смеси новое название и рассказывать всем, какие прекрасные духи она смогла сделать вместе с дочерью.

Считайте, что Оля не пролила ни одной капли, а также что она тщательно встряхивала флаконы после каждого переливания. Учтите, что в маминых флаконах порой не видно, есть ли там жидкость, и потому Оля иногда могла пытаться переливать духи из пустого флакона (в результате, естественно, ничего не переливалось). Вы можете также считать, что после каждого переливания в каждом флаконе каждый тип духов либо полностью отсутствует, либо содержится в объеме не меньшем чем  $10^{-10}$ .

### Формат входных данных

На первой строке входного файла находятся два числа  $N$  и  $M$  — количество флаконов и число типов маминых любимых духов соответственно ( $2 \leq N \leq 100$ ;  $1 \leq M \leq 100$ ). Далее следуют  $N$  строк, на  $i$ -ой из которых находятся два числа — тип  $L_i$  и объем  $V_i$  духов, находившихся изначально в  $i$ -ом флаконе ( $1 \leq L_i \leq M$ ;  $0 \leq V_i \leq 1000$ ). Возможно, что в нескольких флаконах находились духи одного и того же типа; возможно, что какого-то типа вообще не было на полке.

Далее во входном файле следует строка с числом  $K$  — количеством совершённых переливаний ( $1 \leq K \leq 1000$ ). За ней следуют  $K$  строк, на  $k$ -ой из которых находятся три числа  $S_k$ ,  $T_k$  и  $A_k$  — номера флаконов, откуда и куда переливала Оля при  $k$ -ом переливании, и количество перелитой жидкости (в процентах от количества жидкости в  $S_k$ -ом флаконе перед переливанием). Гарантируется, что  $1 \leq S_k, T_k \leq N$ , что  $S_k \neq T_k$ , и что  $0 \leq A_k \leq 100$ . Все числа во входном файле целые.

### Формат выходных данных

В выходной файл выведите  $M$  чисел — процентное содержание всех видов духов (от первого до  $M$ -ого) в первом флаконе после последнего переливания. Выводите результат с точностью не меньше двух знаков после запятой. Гарантируется, что после последнего переливания первый флакон оказался непустым.

### Пример

Входной файл	Выходной файл
3 2 1 100 2 200 1 500 2 3 2 20 2 1 50	60.00 40.00

*Примечание:* В условиях, которые были розданы участника на олимпиаде, из-за ошибки жюри отсутствовало последнее предложение основной части условия («Вы можете также считать, что...»).

<sup>1</sup>... but does anybody really need a  $6\frac{1}{7}$ -point font that is one fourth of the way between Baskerville and Helvetica? — Donald E. Knuth, The Concept of a Meta-Font

## Задача 2. Освещение двора

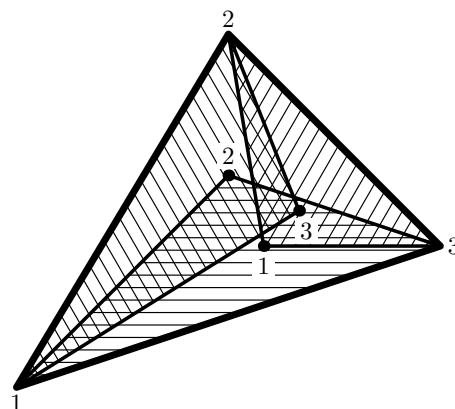
Входной файл	light.in
Выходной файл	light.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_2.pas

Фирма, в которой всё ещё работает ваш друг, собирается расширяться на новые маршруты, и потому приобрела новую площадку для ночного отстоя автобусов. Площадка раньше относилась к военной части, поэтому она имеет необычную форму — форму треугольника, огороженного забором.

Для того, чтобы освещать площадку ночью, на ней надо установить несколько прожекторов. К счастью, у фирмы как раз в наличие есть три регулируемых прожектора, а на территории площадки обнаружили три высоких столба. Было решено на каждый столб повесить по прожектору и отрегулировать их так, чтобы каждый прожектор освещал ровно одну из трёх сторон забора: один прожектор должен освещать одну сторону забора, другой — другую, третий — третью. Никакой прожектор не должен освещать ни миллиметра «чужой» стены.

Конечно, прожекторы должны освещать не только забор, но вообще всю территорию площадки, поэтому возникла проблема: надо определить, какой прожектор на какую сторону забора направить. Зная ваши высокие навыки в решении подобных задач, фирма обратилась к вам за помощью. Напишите программу, которая будет решать эту задачу.

Естественно, каждый прожектор освещает не только стену, но и всю соответствующую часть двора — треугольник с вершиной в месте, где находится столб, на котором висит прожектор, и основанием, совпадающим с соответствующей стороной забора. Будем считать, что стороны этого треугольника тоже освещены прожектором. Тенью от столбов пренебрегайте.



### Формат входных данных

Первая строка входного файла содержит одно число  $T$  — количество тестовых примеров, которые идут дальше ( $1 \leq T \leq 1000$ ).

Далее следуют описания  $T$  примеров. В каждом сначала идут шесть чисел  $x_1, y_1, x_2, y_2, x_3, y_3$  — координаты вершин площадки, после чего идут шесть чисел  $X_1, Y_1, X_2, Y_2, X_3, Y_3$  — координаты столбов. Гарантируется, что все столбы находятся строго внутри площадки. Гарантируется, что площадь площадки строго больше нуля. Гарантируется, что никакие два столба не совпадают. Все координаты во входном файле целые и не превосходят по модулю 800.

### Формат выходных данных

Выведите в выходной файл  $T$  строк по три числа в каждой: для каждого примера выведите номер стороны забора, на которую должен светить прожектор, находящийся на первом столбе, потом номер стороны, на которую должен светить прожектор со второго столба, и наконец номер стороны, на которую должен светить прожектор с третьего столба.

Первая сторона — та, которая соединяет вершины  $(x_1, y_1)$  и  $(x_2, y_2)$ , вторая —  $(x_2, y_2)$  и  $(x_3, y_3)$ , третья —  $(x_3, y_3)$  и  $(x_1, y_1)$ .

Если решения не существует, выведите в соответствующую строку выходного файла три минуса единицы: '-1 -1 -1'.

### Пример

Входной файл	Выходной файл
2	2 3 1
0 0 6 10 12 4	3 2 1
7 4 6 6 8 5	
-10 -10 0 10 10 -10	
0 1 0 -1 1 1	

*Примечание:* Первый пример соответствует рисунку.

*Примечание:* Среди тестов будут такие, в которых  $T \leq 10$ ; суммарная стоимость таких тестов будет 70 баллов.

## Задача 3. Совершенно несчастливые билеты

Входной файл	tickets.in
Выходной файл	tickets.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_3.pas

Миша часто ездит в маршрутках. Миша законопослушный, поэтому он всегда покупает билет. Каждый билет в маршрутке имеет номер —  $2N$ -значное десятичное число, возможно, с ведущими нулями. После покупки билета Миша всегда проверяет, счастливый ли достался ему билет. Счастливым Миша считает такой билет, у которого сумма первых  $N$  цифр номера равна сумме последних  $N$  цифр. Если билет оказывается несчастливым, Миша ищет расстояние до ближайшего счастливого, т.е. минимальное число  $x$  такое, что если к номеру билета, полученного Мишей, прибавить или отнять  $x$  (при этом, разумеется, полученный номер должен быть корректным номером билета, т.е. должен быть не меньше нуля и не больше  $10^{2N} - 1$ ), то получившийся номер должен быть номером счастливого билета.

Билеты, у которых это расстояние максимально среди всех возможных, Миша называет совершенно несчастливыми. Мише очень интересно, сколько всего существует совершенно несчастливых билетов и какие номера у этих билетов. Но так как Миша плохо учился в школе, он не знает, как решать такую задачу, поэтому он обратился за помощью к вашему другу, специалисту по маршруткам. Он же перенаправил Мишу к вам.

### Формат входных данных

Во входном файле содержится единственное число  $N$  ( $1 \leq N \leq 100\,000$ ) — половина длины номера билетов.

### Формат выходных данных

В первой строке выходного выведите одно число  $k$  — количество совершенно несчастливых билетов. В последующих  $k$  строках выведите номера билетов в порядке возрастания. Номера нужно выводить с ведущими нулями, если таковые есть.

### Пример

Входной файл	Выходной файл
2	4 0050 0051 9948 9949

*Примечание:* В примере расстояние от билета с номером 0050 до ближайшего счастливого равно 50: если из этого номера вычесть 50, получится 0000 — счастливый номер. Аналогично, от билета 0051 расстояние до ближайшего счастливого тоже равно 50: если к номеру прибавить 50, получится 0101. У билетов 9948 и 9949 расстояние до ближайшего счастливого также равно 50; других билетов с таким расстоянием нет. Билетов с большим расстоянием тоже нет, поэтому эти четыре билета и только они являются совершенно несчастливыми.

## Задача 4. BubbleGum

Входной файл	bubble.in
Выходной файл	bubble.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_4.pas

Мальчик Влад недавно побывал в Японии и привёз оттуда новую жевательную резинку. Вернувшись в университет после поездки, на первой же паре Влад раздал жвачку всем своим  $(N - 1)$  однокурсникам и взял одну себе. Дождавшись момента, когда лектор отвернулся к доске, на счёт «три-четыре» все  $N$  студентов дружно начали надувать пузыри. Известно, что  $i$ -й студент надувает пузырь до максимально возможного размера за время  $t_i$ , после чего пузырь мгновенно лопается, и студент начинает надувать пузырь заново с той же скоростью.

Всё это время преподаватель настолько увлечён тонкостями квантового математического анализа, что не слышит ничего происходящего в аудитории. И только когда все  $N$  пузырей лопнут *одновременно*, преподаватель услышит шум и обернётся. И уж тогда студентам достанется, а больше всех тому, кто принёс на пару  $N$  жевательных резинок.

Определите, сколько времени студенты смогут наслаждаться надуванием пузырей, не замечаемые преподавателем.

Например, если  $N = 2$ ,  $t_1 = 2$ ,  $t_2 = 3$ , то будет происходить следующее:

Первый студент надувает пузырь с момента времени  $t = 0$  до момента времени  $t = 2$ , потом пузырь лопается, и он надувает пузырь заново — с момента времени  $t = 2$  до момента времени  $t = 4$ , а потом ещё раз — с момента времени  $t = 4$  до  $t = 6$ .

Второй студент надувает пузырь с  $t = 0$  до  $t = 3$  и ещё раз с  $t = 3$  до  $t = 6$ .

В момент  $t = 6$  пузыри лопаются одновременно у обоих студентов, преподаватель оборачивается и говорит: «Всё, Влад! Ты меня достал!».

### Формат входных данных

На первой строке входного файла находится одно целое число  $N$  — количество студентов ( $1 \leq N \leq 10\,000$ ). На второй строке входного файла находятся  $N$  целых чисел  $t_1, t_2, \dots, t_N$ . Гарантируется, что  $1 \leq t_i \leq 1000$ .

### Формат выходных данных

Выведите в выходной файл одно число — время, в течение которого студенты во главе с Владом могут наслаждаться безнаказанным надуванием пузырей.

### Пример

Входной файл	Выходной файл
2 2 3	6
1 1	1
2 16 1	16
3 627 182 85	9699690

## Задача 5. Шоколадки

Входной файл	choco.in
Выходной файл	choco.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_5.pas

У мальчика Васи есть  $N$  шоколадок (возможно, разного веса). Вася пригласил к себе в гости  $K$  своих друзей и хочет подарить им шоколадки. Чтобы никому из друзей не было обидно, Вася решил раздать шоколадки так, чтобы каждому другу досталось одно и то же количество шоколада (т.е. суммарный вес шоколадок, доставшихся каждому другу, должен быть одинаковым). Вася может раздать все свои шоколадки, может раздать лишь часть, но, поскольку он — очень гостеприимный мальчик, он не хочет оставлять друзей совсем без шоколада (т.е. сумма весов шоколадок, доставшихся каждому другу, должна быть строго положительной). Все шоколадки красиво упакованы, т.е. делить их на части нельзя.

Определите, сколько у Васи есть способов раздать шоколад своим друзьям. Два способа считайте различными тогда и только тогда, когда существует шоколадка, которая в одном способе досталась некоторому другу, а в другом — другому другу или вовсе не была отдана друзьям.

### Формат входных данных

В первой строке входного файла находятся два натуральных числа  $N$  и  $K$  ( $1 \leq N \leq 15$ ,  $1 \leq K \leq 15$ ) — количество шоколадок у Васи и количество друзей, которых Вася пригласил в гости. Во второй строке содержатся  $N$  натуральных чисел — веса шоколадок. Ни один из весов не превосходит 1000.

### Формат выходных данных

Выведите в выходной файл одно число — количество способов раздать шоколадки друзьям.

### Пример

Входной файл	Выходной файл
5 4 1 2 1 1 1	24
3 2 1 1 2	4

*Примечание:* Во втором примере возможные распределения шоколадок следующие:

- 1) Первому другу дать шоколадку номер 1, второму — номер 2;
- 2) Первому другу дать шоколадку номер 2, второму — номер 1;
- 3) Первому другу дать шоколадку номер 3, второму — шоколадки номер 1 и 2;
- 4) Первому другу дать шоколадки номер 1 и 2, второму — номер 3.

## Задача 6. Два капитана

Входной файл	treasure.in
Выходной файл	treasure.out
Ограничение по времени	1 сек
Максимальный балл за задачу	100
Пример названия программы	p99_6.pas

Капитаны Флинт и Джек Воробей нашли клад и хотят поделить его. Клад находится в шкатулке и состоит из чётного числа драгоценных камней. Капитан Флинт оценил  $i$ -ый камень в  $a_i$  пиастров, а Джек Воробей — в  $b_i$  долларов. Теперь они действуют следующим образом. Джек Воробей достаёт из шкатулки два камня, после чего Флинт забирает себе один из них (естественно, тот, у которого больше  $a_i$ ). Оставшийся камень достаётся Воробью. После этого Джек Воробей достаёт ещё пару камней, и так далее: каждым ходом Воробей достаёт из шкатулки два камня, Флинт забирает себе камень с большим  $a_i$ , оставшийся камень достаётся Воробью.

Джек Воробей знает все  $a_i$ , все  $b_i$ , а также может, доставая очередные два камня, подглядеть в шкатулку и выбрать, какие именно камни надо доставать. Помогите ему действовать так, чтобы доля Воробья была максимально возможной (т.е. чтобы сумма  $b_i$  полученных Воробьём камней была как можно больше).

По сравнению с камнями шкатулка ничего не стоит, поэтому её можно не учитывать при дележе.

### Формат входных данных

Первая строка входного файла содержит одно целое число  $N$  — количество камней в кладе. Гарантируется, что  $N$  чётное и что  $2 \leq N \leq 5000$ . Далее следуют две строки по  $N$  целых чисел в каждой: сначала заданы все  $a_i$ , потом — все  $b_i$ . Гарантируется, что все  $a_i$  различны (т.е. что действия Флинта всегда однозначно определены). Гарантируется, что все  $a_i$  и все  $b_i$  положительны и не превосходят 400 000.

### Формат выходных данных

В выходной файл выведите  $N/2$  строк по два числа в каждой — пары камней, в том порядке, как их должен доставать из шкатулки Джек Воробей. Камни нумеруются начиная с 1.

Числа в пределах каждой пары можете выводить в произвольном порядке. Если есть несколько оптимальных решений, выводите любое.

### Пример

Входной файл	Выходной файл
6	5 1
6 10 11 18 5 14	2 3
1 7 6 12 15 16	6 4
6	3 1
6 44 2 43 7 48	5 4
6 44 2 43 7 48	2 6

*Примечание:* Среди тестов будут такие, в которых каждый камень оба капитана оценивают одинаково:  $a_i = b_i$  для каждого  $i$  (как во втором тесте из примера); суммарная стоимость таких тестов будет 40 баллов.