

## Задача 1. Расписание маршруток

Входной файл	schedule.in
Выходной файл	schedule.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	p99_1.pas

Оценив последние успехи вашего друга в экономическом отделе компании, владеющей сетью маршрутных такси, директор повысил его до главного диспетчера. Теперь друг указывает, на какой маршрут должна выходить та или иная маршрутка.

В автопарке компании есть  $n$  маршруток,  $i$ -ая маршрутка номинально вмещает  $a_i$  пассажиров. По договору с департаментом транспорта города компания обязана обслуживать  $m$  маршрутов. Накопленная статистика говорит, что оптимальнее всего, если  $j$ -ый маршрут обслуживает такси номинальной вместимостью  $b_j$  пассажиров. Каждая маршрутка приписывается не более чем к одному маршруту, каждому маршруту приписывается не более одной маршрутки.

Разумеется, каждый уважающий себя диспетчер при назначении маршруток на маршруты старается минимизировать потери, которые бывают следующими:

- если  $i$ -ая маршрутка обслуживает  $j$ -ый маршрут, то компания теряет  $|a_i - b_j|$  у.е., так как чем меньше заполнено такси, тем больше не используются его возможности, а чем больше переполнено такси, тем чаще его приходится ремонтировать;
- от каждой простаивающей маршрутки, то есть такой, которой не назначен ни один маршрут, компания несет убыток  $p$  у.е.;
- компании приходится платить штраф  $q$  у.е. департаменту транспорта за каждый не обслуживаемый маршрут.

В очередной раз вы хотите помочь другу и написать для него программу, облегчающую ему работу.

### Формат входных данных

В первой строке входного файла находятся четыре целых числа —  $n, m, p, q$  ( $1 \leq n, m \leq 10^3, 0 \leq p, q \leq 10^4$ ).

Во второй строке через пробел указаны целые числа  $a_1, \dots, a_n$  ( $1 \leq a_i \leq 10^4$ ).

В третьей строке через пробел указаны целые числа  $b_1, \dots, b_m$  ( $1 \leq b_j \leq 10^4$ ).

### Формат выходных данных

Выведите единственное число — минимально возможные потери компании.

### Пример

Входной файл	Выходной файл
2 2 100 100 22 12 11 20	3
2 1 100 500 13 13 13	100

*Примечание:* В примере 1 первая маршрутка назначена на второй маршрут с потерями  $|22 - 20| = 2$  у.е., вторая маршрутка назначена на первый маршрут с потерями  $|12 - 11| = 1$  у.е.. Итого: потери 3 у.е..

В примере 2 одна из маршруток назначается на единственный маршрут с нулевым штрафом, а вторая вынуждена простаивать. Итого: потери 100 у.е.

## Задача 2. Праздничные дни

Входной файл	holidays.in
Выходной файл	holidays.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МиБ
Максимальный балл за задачу	100
Пример названия программы	p99_2.pas

В Тридевятиом царстве царь был любителем разных заморских традиций. Как прознает, что в другом царстве есть какой-то обычай, сразу думает, как бы его к тридевятым реалиям приспособить.

Вот неделю назад вернулось посольство из Тридесятого царства. И главный посол доложил царю: дескать, придумал Тридесятый царь следующую вещь. Чтобы как-то зарегулировать гуляния народные, повелел он указать определенные дни, и в эти дни устраивать широкие гуляния, а в остальные дни массовые сборища запретить. И с тех пор жизнь в Тридесятом царстве стала прекрасной: гулять так гулять, работать так работать, и все строго по цареву указу.

Понравилась мысль такая царю Тридевятого царства. Подумал он ввести и у себя такие порядки. Собрал царь советников своих, и говорит: подготовьте мне список дней, в которые гулять можно. Только не на год, а на  $N$  дней вперед — посмотрим, дескать, что получится; понравится — сделаем круглогодичным.

И вот вчера принесли советники царю список. Но вот незадача: каждый советник свой список приговил, да еще и обоснование предложил, какой праздник в какой из этих дней надо отмечать. И у всех советников праздники важные, но у всех — разные! Царь думал-думал и решил: а возьмем их все — объединим предложения советников! Если какой-то день есть в списке хотя бы одного советника, то объявим этот день праздничным, и пускай народ гуляет! Глядишь, и не будет недовольных.

Только одна проблема осталась: некоторые дни оказались в списках сразу у нескольких советников. Но царь и тут нашел выход: перенесем некоторые праздники на более поздние дни, так, чтобы в каждый день получался только один праздник, и переносы были бы как можно короче.

Пусть, например, четыре советника сразу предложили сделать некоторый день (пускай день 5) праздничным. Тогда перенесем три из этих четырех праздников на дни 6, 7 и 8 — так, что праздничными будут дни с 5 по 8 включительно. А если оказывается, что, например, день 7 тоже предложен в качестве праздничного кем-нибудь из советников, то перенесем этот праздник еще дальше — на день 9.

Напишите программу, которая, зная предложения советников, определит, какие дни будут праздничными, а какие нет. Не забывайте, что праздники можно переносить только на более поздние дни; на более ранние переносить нельзя.

### Формат входных данных

На первой строке входного файла находится одно число  $N$  — количество дней, на которые царь хочет произвести планировку праздников.

На второй строке входного файла находятся  $N$  неотрицательных целых чисел — для каждого дня указано, сколько советников предложили считать его праздничным.

Гарантируется, что  $1 \leq N \leq 100\,000$ , и что сумма всех чисел во второй строке входного файла не превосходит 100 000.

### Формат выходных данных

В выходной файл выведите одну строку, состоящую из символов '+' или '-' (без апострофов). Символом '+' обозначайте праздничный день, '-' — непраздничный. Выведите как минимум  $N$  символов — по одному для каждого из дней, на которые проводится планирование. Но если праздники приходится переносить на дни после  $N$ -ого (что допустимо), то выведите больше символов — до последнего праздничного дня.

Символы разделяйте пробелами.

### Пример

Входной файл	Выходной файл
5 0 3 0 0 0	- + + + -
10 0 4 0 2 0 0 0 0 1 0	- + + + + + - + -
3 0 3 0	- + + +

## Задача 3. Выставка ноутбуков

Входной файл	http.in
Выходной файл	http.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	p99_3.pas

Сеть компьютерных салонов «ХТТП» представлена в городе Н. двумя магазинами. Руководство Н-ского отделения сети решило реорганизовать витрины, на которых представлены ноутбуки. В каждом из двух магазинов на витрине должны быть представлены  $N$  моделей ноутбуков, выставленные в ряд от касс вглубь помещения магазина. Маркетологи каждого из магазинов уже определили порядок, в котором на витрине должны быть расположены эти модели (эти порядки в двух магазинах, конечно же, могут быть разными).

На витрины надо выставлять специальные, выставочные, образцы ноутбуков, с соответствующим программным обеспечением, показывающим рекламу, и т.д. В распоряжении администрации компьютерных салонов есть две версии специализированного ПО: работающие под управлением операционных систем Windows и Linux. Соответственно, каждый из выставочных образцов ноутбуков должен иметь предустановленной ровно одну из этих систем.

Для снижения затрат было принято решение закупить по два идентичных экземпляра каждой модели ноутбуков (т.е. с одинаковыми предустановленными операционными системами), отправить по одному экземпляру в каждый из магазинов сети, и расставить их на витрине в соответствии с порядком, определенным маркетологами того магазина.

Но при этом возникла проблема. По требованиям Федеральной антимонопольной службы, компьютерные салоны не должны предоставлять преимущества ни одной из операционных систем. Начальство сети «ХТТП» знает, как проходит проверка на соответствие этой норме законодательства. Инспектор антимонопольной службы идет по магазину начиная от касс вдоль витрины с ноутбуками, считает отдельно количество встреченных ноутбуков с Windows и Linux, а также модуль разности этих чисел (т.е. на сколько ноутбуков с одной системой он встретил больше, чем ноутбуков с другой системой). В некоторый момент он останавливается и говорит: «Ага!». Это значит: слишком у вас большой дисбаланс между операционными системами, поэтому платите штраф. Размер штрафа пропорционален разнице (взятой по модулю) количества ноутбуков с разными системами, которые увидел инспектор.

Естественно, руководство сети не в состоянии предсказать, в какой из магазинов пойдет инспектор, а также сколько ноутбуков он просмотрит. Тем не менее, они хотят минимизировать штраф, который им будет выписан *в худшем случае*. Помогите им.

Например, пусть  $N = 5$ : в магазинах должны быть выставлены пять моделей ноутбуков. Будем нумеровать модели ноутбуков от 1 до 5. Пусть в первом магазине маркетологи определили, что оптимальный порядок ноутбуков следующий (от касс вглубь зала):

2 4 1 3 5,

а во втором магазине —

3 1 2 5 4.

Тогда, если закупить ноутбуки моделей 1, 3 и 4 с операционной системой Windows, а 2 и 5 — с Linux, то порядок операционных систем в магазинах будет следующий (от касс вглубь зала):

L W W W L,  
W W L L W.

Тогда, если, например, инспектор придет в первый магазин и просмотрит первые четыре ноутбука, то он скажет: «Ага!», и выпишет штраф за то, что он увидел Windows-ноутбуков на два больше, чем Linux. Аналогичный результат будет, если он придет во второй магазин и просмотрит только первые два ноутбука.

А если закупить ноутбуки 2 и 3 с системой Linux, а 1, 4, 5 — с Windows, то порядок операционных систем будет следующий:

L W W L W,  
L W L W W,

и в какой бы магазин не пришел инспектор, и сколько бы ноутбуков он не посмотрел, разница Windows- и Linux-ноутбуков не будет превосходить по модулю 1, и это и будет оптимальным вариантом для руководства сети. (Напомним, что инспектор всегда начинает осмотр магазина от касс и идет вглубь магазина вдоль ряда с ноутбуками).

*Продолжение условия на следующей странице*

### Формат входных данных

В первой строке входного файла записано одно число  $N$  ( $1 \leq N \leq 10^5$ ) — количество моделей ноутбуков, которые должны быть представлены на витрине. Модели ноутбуков нумеруются от 1 до  $N$ .

Далее следуют две строки по  $N$  чисел в каждой — порядки моделей ноутбуков на витрине, определенные маркетологами первого и второго магазина, от касс вглубь зала. Гарантируется, что порядки корректные, т.е. что в каждой из этих строк все числа находятся в интервале от 1 до  $N$  и никакое из чисел не встречается в одной строке дважды.

### Формат выходных данных

В выходной файл выведите строку из  $N$  символов, описывающих необходимую конфигурацию ноутбуков. А именно,  $i$ -ый символ должен быть 'W' (без кавычек), если  $i$ -ую модель ноутбуков надо закупать с предустановленной системой Windows, и 'L' для Linux.

Если есть несколько оптимальных решений, выведите любое.

### Пример

<i>Входной файл</i>	<i>Выходной файл</i>
5 2 4 1 3 5 3 1 2 5 4	WLLWW
5 1 4 2 3 5 5 1 3 4 2	WLWLL

## Задача 4. Тестирование программы

Входной файл	testing.in
Выходной файл	testing.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МиБ
Максимальный балл за задачу	100
Пример названия программы	p99_4.pas

Вчера у Васи был счастливый день: он наконец дописал программу всей своей жизни! И, недолго думая, он тут же запустил ее на Самом Главном Тесте.

Программа Васи очень сложная, и потому работает она долго. Поэтому Вася пошел спать, а наутро, сегодня, обнаружил, что программа, вместо того, чтобы посчитать нужный Ответ, вылетела с непонятной ошибкой.

Вася понял, что напрасно он не тестировал программу. Ведь Самый Главный Тест очень сложный — в нем есть  $N$  отдельных подзадач, и каждую из них надо решить. Конечно, надо было бы начать тестирование с меньшего количества подзадач, но ведь Вася — очень умный программист! И он абсолютно уверен, что его программа корректно решила все подзадачи, кроме какой-то одной. Вот только Вася не знает, какой именно.

Вася может изменять Самый Главный Тест, отключая в нем те или иные подзадачи. Он надеется, что, если среди отключенных будет та подзадача, на которой его программа не работает, то получившийся тест его программа спокойно пройдет. Но вот проблема: программа работает долго, а подзадач много, и потому, если отключать задачи по одной, то придется очень долго искать нужную. А Вася очень хочет узнать, где ошибка, уже завтра!

К счастью, у Васи в распоряжении есть много компьютеров. Он может на некоторых из них запустить свою программу на каком-то тесте (т.е. на Самом Главном Тесте с некоторыми отключенными подзадачами), а на завтра посмотреть, какие программы завершились успешно, а какие нет, — и по результатам понять, какая подзадача создавала проблемы. Помогите Васе подобрать тесты для каждого из компьютеров, т.е. объясните ему, какие подзадачи в каком тесте он должен отключить, так, чтобы на завтра, узнав результаты работы программы на этих тестах, он смог бы уверенно определить, с какой подзадачей у него возникают проблемы (естественно, считая, что такая подзадача только одна, ведь Вася — очень умный программист!)

Учтите, что Васе не хочется делать лишнюю работу по запуску программ и определению их результата, поэтому он хочет минимизировать количество запусков (т.е. фактически количество компьютеров, ведь его программа настолько сложная, что на одном компьютере можно запустить только один экземпляр программы).

### Формат входных данных

В первой и единственной строке входного файла находится одно целое число  $N$  — количество подзадач в Самом Главном Тесте ( $1 \leq N \leq 10^5$ ).

### Формат выходных данных

В первой строке выходного файла выведите минимальное необходимое количество компьютеров  $M$ . В последующих  $M$  строках выведите информацию о том, на каком тесте надо запускать программу на каком компьютере. А именно, в  $i$ -ую из них выведите последовательность чисел  $k_i, b_{i1}, b_{i2}, \dots, b_{ik_i}$ , где  $k_i$  — количество подзадач, которые надо отключить в тесте, на котором будет работать программа на  $i$ -ом компьютере, а  $b_{ij}$  ( $1 \leq j \leq k_i, 1 \leq b_{ij} \leq N$ ) — номера подзадач, которые надо отключать. Числа  $b_{ij}$  должны быть различны для каждого фиксированного  $i$ .

Подзадачи нумеруются от 1 до  $N$ .

В пределах одной строки подзадачи можете выводить в любом порядке. Если есть несколько оптимальных решений, выведите любое.

*Пример и примечание на следующей странице*

**Пример**

<i>Входной файл</i>	<i>Выходной файл</i>
5	3 3 1 3 5 3 1 2 5 4 1 2 3 4

*Примечание:* В примере:

- если ошибка возникала на первой подзадаче, то на всех трех компьютерах программа отработает верно, т.к. первая подзадача во всех тестах отключена;
- если ошибка возникала на второй подзадаче, то ошибочно сработает только первый компьютер;
- если ошибка возникала на третьей подзадаче, то ошибочно сработает только второй компьютер;
- если ошибка возникала на четвертой подзадаче, то ошибочно сработают первый и второй компьютеры;
- если ошибка возникала на пятой подзадаче, то ошибочно сработает только третий компьютер.

Поскольку во всех пяти вариантах множества ошибочно сработавших компьютеров различны, то Вася наутро, узнав, на каких компьютерах программа сработала корректно, а на каких — нет, сможет однозначно определить ошибочную подзадачу.

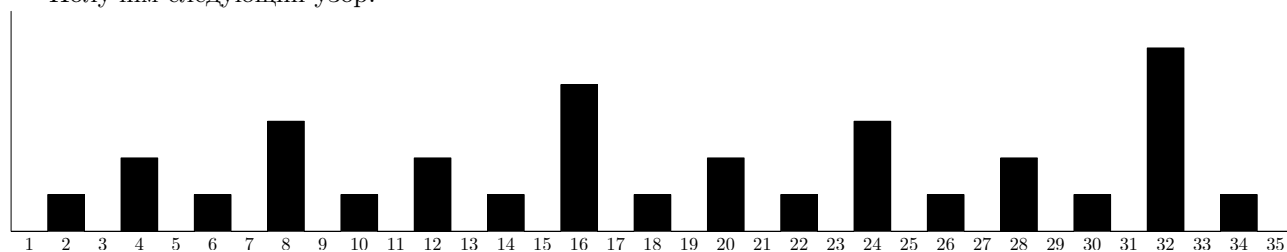
## Задача 5. Раскраска автобусов

Входной файл	paint.in
Выходной файл	paint.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	p99_5.pas

Ваш друг по-прежнему работает главным экономистом в компании, владеющей сетью маршруток. В связи с новыми веяниями по введению корпоративной раскраски на транспортных средствах, компания решила окрасить все свои маршрутки в новом стиле. Директор компании своим личным распоряжением приказал принять следующий способ окраски. Сначала все автобусы окрашиваются фоновой краской, поверх которой наносится серый узор.

Узор создается следующим образом. Если бы борт маршрутки был бы бесконечным в две стороны: вправо и вверх, — то узор рисовали бы так. Разделим борт на бесконечное количество вертикальных полос одинаковой (единичной) ширины, занумеруем их слева направо, начиная с единицы. Полосы с нечетными номерами красить серой краской вообще не будем. Полосы, номера которых делятся на два, но не делятся на четыре, закрасим снизу до высоты, равной единице длины (т.е. образуется серый квадрат). Полосы, номера которых делятся на четыре, но не делятся на восемь, закрасим снизу до высоты, равной двум единицам длины; полосы, номера которых делятся на восемь, но не делятся на 16 — до высоты, равной 3; номера которых делятся на 16, но не делятся на 32 — до высоты, равной 4, и т.д.

Получим следующий узор:



Естественно, у реальных маршруток ширина борта ограничена (высоту мы для простоты будем считать неограниченной). Можно было бы на каждой маршрутке рисовать начало такого узора, но это не интересно — поэтому было решено для каждой маршрутки выбрать два числа,  $l$  и  $r$ , и нарисовать на борту фрагмент этого узора с  $l$ -ого столбика по  $r$ -ый включительно. Определите, сколько на это уйдет серой краски, считая, что единица краски уходит на единичный квадрат узора.

### Формат входных данных

Во входном файле находятся два числа —  $l$  и  $r$ . Гарантируется, что  $1 \leq l \leq r \leq 10^{18}$ .

### Формат выходных данных

Выведите в выходной файл одно число — общую площадь фрагмента узора между  $l$ -ым и  $r$ -м столбиками включительно.

### Пример

Входной файл	Выходной файл
5 10	5

*Примечание:* В примере используются столбики с 5 по 10 включительно. Их площади — соответственно 0, 1, 0, 3, 0, 1 единичных квадратов; соответственно, суммарная их площадь равна 5.

*Примечание:* Среди тестов будут такие, в которых  $r \leq 10^5$ , общей стоимостью 40 баллов, и тесты с  $r > 10^5$ , но  $r - l \leq 10^5$ , общей стоимостью еще 20 баллов.

## Задача 6. Руммикуб

Входной файл	rummikub.in
Выходной файл	rummikub.out
Ограничение по времени	1 сек
Ограничение по памяти	256 МБ
Максимальный балл за задачу	100
Пример названия программы	r99_6.pas

В игре «Руммикуб» используются фишки, которые бывают четырех различных цветов, и на каждой из которых написано одно натуральное число от 1 до 13. Для каждого числа и для каждого цвета в наборе фишек есть ровно две соответствующие фишки, т.е. всего в наборе  $8 \cdot 13 = 104$  фишки.

Число, написанное на фишке, будем называть ее *достоинством*; цвета будем обозначать латинскими буквами А, В, С и D, и каждую фишку будем обозначать, записывая сначала ее цвет, а потом — ее достоинство. Например, C12 — это фишка цвета С и достоинством 12.

*Комбинацией* в игре называется набор из как минимум трех фишек, удовлетворяющий любому из следующих условий:

- Достоинства всех фишек одинаковы, а цвета — попарно различны; или
- Цвета всех фишек одинаковы, а достоинства являются последовательными натуральными числами.

Например, следующие наборы фишек являются комбинациями:

- C12, A12, B12;
- C12, A12, B12, D12;
- C5, C6, C7;
- A3, A4, A5, A6, A7.

При этом следующие наборы *не являются* комбинациями:

- A3, B3 (слишком мало фишек);
- A3, B3, C3, D3, B3 (цвета повторяются);
- A3, A4, A4, A5, A6 (достоинства повторяются);
- A3, A4, A6, A7, A8 (число 5 пропущено).

Одна из основных задач в руммикубе состоит в том, чтобы данный набор фишек распределить на комбинации так, чтобы каждая фишка входила ровно в одну комбинацию. Напишите программу, которая будет это делать.

### Формат входных данных

В первой строке входного файла находится одно натуральное число  $K$  — количество фишек. Далее следуют  $K$  строк, на каждой из которых находится описание одной фишки — цвет и достоинство.

Гарантируется, что эти фишки являются корректным подмножеством фишек из некоторого комплекта для игры в руммикуб; т.е. гарантируется, что каждый цвет — это латинская заглавная буква от А до D, что каждое достоинство — это натуральное число, не превосходящее 13, и что для каждой пары (цвет, достоинство) в наборе есть не более двух таких фишек.

### Формат выходных данных

Если данный набор фишек можно разбить на комбинации так, чтобы каждая фишка входила ровно в одну комбинацию, то в первую строку выходного файла выведите одно число  $M$  — количество комбинаций в вашем решении. Далее выведите  $M$  строк, в  $i$ -ой из которых выведите  $i$ -ую комбинацию. А именно, сначала выведите количество фишек в комбинации, а потом сами фишки, разделенные между собой и отделенные от количества фишек пробелами. В пределах каждой комбинации фишки можете выводить в произвольном порядке; комбинации также можете выводить в произвольном порядке.

Если есть несколько решений, выведите любое. Если решений нет, то выведите в выходной файл одно число -1.

*Пример на следующей странице*



**Пример**

<i>Входной файл</i>	<i>Выходной файл</i>
3 A2 A3 A5	-1
3 A2 A4 A3	1 3 A2 A4 A3
7 A12 A13 A13 B13 C13 D13 A11	2 3 A11 A12 A13 4 A13 B13 C13 D13