

Задача А. Деньги

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

У вас имеется неограниченное количество монеток N заданных достоинств. Вам требуется определить, можно ли с их помощью разменять заданные K сумм денег.

Формат входных данных

В первой строке число N , далее во второй строке N чисел, задающих достоинства монеток. В третьей строке — число K , далее в четвёртой K чисел, определяющих размеры сумм. Все заданные числа натуральные и не превосходят 1000.

Формат выходных данных

В единственной строке выведите K чисел — для каждой суммы 0, если её разменять нельзя, и 1, если можно.

Пример

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|-------------------------|--------------------------|
| 2 | 1 1 0 1 1 |
| 3 5 | |
| 5 | |
| 3 6 7 11 12 | |

Задача В. Мне это больше не нужно!

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Один отважный путешественник отправился облетать шар земной на шаре воздушном. К сожалению, его воздушный шар подхватило бурей. Шар начал падать, и путешественник приготовился было умирать, но, к счастью, на горизонте показался остров. Остров был маленький, однако стало ясно, что если выкинуть из корзины вещи, то можно до него долететь, и даже перелететь. Известно, что если путешественник выкинет вещи суммарным весом больше, чем w , то он перелетит за край острова и упадёт в бушующее море, где его ждёт неминуемая смерть. Естественно, что его не устраивает такой исход, и потому он хочет выкинуть вещей, с одной стороны, как можно большего суммарного веса (чтобы гарантированно оказаться на острове), с другой стороны, этот вес ни в коем случае не должен превосходить w . Нужно написать программу, которая бы решила бы, какие вещи нужно выкидывать. Помогите отважному путешественнику!

Формат входных данных

В первой строке заданы два целых числа: n (число вещей в корзине воздушного шара, $1 \leq n \leq 20$) и w (предельный вес, $1 \leq w \leq 10^9$). Во второй строке записаны n целых чисел, разделённые пробелами, каждое из которых описывает вес некоторой вещи, находящейся в корзине воздушного шара. Каждое из этих чисел лежит в пределах от 1 до 10^6 .

Формат выходных данных

В первой строке необходимо выдать максимально возможное значение суммарного веса w_{max} . Во второй строке необходимо вывести список номеров вещей, взяв которые, можно достигнуть такого веса. Номера должны выводиться через пробел. Если оптимальных решений несколько, разрешается выводить любое из них.

Пример

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|-------------------------|--------------------------|
| 5 13 | 12 |
| 3 5 6 9 9 | 1 5 |

Задача С. Наилучшее приближение

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Вам даны N целых чисел. Ваша задача — вставить ровно по одному знаку «+» или «-» между каждой парой соседних таким образом, чтобы сделать значение получившегося выражения максимально близким к заданному числу A .

Формат входных данных

Первая строка содержит два целых числа: N ($1 \leq N \leq 10\,000$) и A , которое по модулю не превосходит 10 000. Далее следуют N строк, в каждой из которых содержится ровно одно целое число X_i , не превосходящее по модулю 10 000. Кроме того, гарантируется, что сумма абсолютных величин всех N чисел также не превосходит 10 000.

Формат выходных данных

В первой строке необходимо вывести значение получившегося выражения (которое должно быть настолько близко к A , насколько это возможно). Во второй строке необходимо вывести само выражение, дающее такое значение, в форме $X_1[+|-]X_2[+|-]\dots X_{N-1}[+|-]X_N$. Если оптимальных решений несколько, то разрешается выводить любое из них.

Пример

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

Задача D. Делёж монет

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В этой задаче требуется разделить K монет поровну на троих, или выяснить, что это невозможно.

Формат входных данных

В первой строке задано число K — количество монет ($1 \leq K \leq 20$). Во второй строке даны K целых чисел A_1, A_2, \dots, A_K через пробел — достоинства монет ($1 \leq A_i \leq 100$).

Формат выходных данных

В первой строке выведите «YES», если справедливый делёж монет возможен, и «NO», если разделить монеты поровну не удастся. Кроме того, если ответ положительный, в следующих трёх строках выведите через пробел наборы монет, на которые нужно разделить исходный набор, чтобы сумма монет в каждом наборе оказалась одинаковой. Если возможных решений несколько, выведите любое из них.

Примеры

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|-------------------------|--------------------------|
| 3 1 1 1 | YES 1 1 1 |
| 2 1 10 | NO |
| 5 1 2 3 4 5 | YES 1 4 2 3 5 |
| 6 1 2 4 8 16 32 | NO |

Задача Е. Билеты в кино

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В этой задаче вам предлагается помочь кассиру, продающему билеты в кино. Осталось два свободных ряда, один за другим, по m мест в каждом. Места в каждом ряду нумеруются слева направо числами от 1 до m . В очереди стоят люди группами по a_i человек. Каждую группу можно посадить в один из рядов подряд, либо, если a_i чётное, можно посадить её в два ряда на места с одинаковыми номерами.

Кассир в раздумьи: удастся ли ему посадить все группы, соблюдая эти требования? Помогите ему, найдя минимальную длину ряда m , при которой можно посадить все группы, соблюдая их.

Формат входных данных

В первой строке записано целое число n — количество групп ($1 \leq n \leq 1000$). Во второй строке заданы через пробел n целых положительных чисел a_1, a_2, \dots, a_n ; здесь a_i — количество людей в i -й группе. Сумма всех a_i не превосходит 100 000.

Формат выходных данных

Выведите одно целое число — минимальную длину одного ряда, при которой получится посадить все n групп.

Примеры

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|-------------------------|--------------------------|
| 4 1 2 3 4 | 5 |
| 3 12 5 3 | 11 |

Задача F. Две сумки

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Ральф купил в магазине n вещей. Вещи пронумерованы целыми числами от 1 до n . Вещь с номером i имеет объём v_i и вес w_i . Осталось разложить вещи по сумкам, чтобы унести покупки домой.

У Ральфа есть две сумки. Первая сумка — прочная: в неё можно положить вещи любого веса, но их суммарный объём должен быть не больше x . Вторая сумка — вместительная: в неё можно положить вещи любого объёма, но их суммарный вес должен быть не больше y .

Помогите Ральфу, если это возможно, разложить вещи по двум сумкам так, чтобы:

- каждая вещь оказалась в одной из двух сумок,
- суммарный объём вещей в первой сумке был не больше x ,
- суммарный вес вещей во второй сумке был не больше y .

Формат входных данных

В первой строке заданы три целых числа n , x и y — количество вещей, ограничение на объём в первой сумке и ограничение на вес во второй сумке ($1 \leq n \leq 500$, $1 \leq x, y \leq 250\,000$). Во второй строке записаны n целых чисел v_1, v_2, \dots, v_n — объёмы вещей ($1 \leq v_i \leq 250\,000$). В третьей строке записаны n целых чисел w_1, w_2, \dots, w_n — веса вещей ($1 \leq w_i \leq 250\,000$).

Формат выходных данных

Если решение существует, выведите строку из n маленьких английских букв. Буква на i -й позиции определяет, в какую сумку положить i -ю вещь. Буква «x» обозначает первую сумку (прочную), а буква «y» — вторую сумку (вместительную). Суммарный объём вещей в первой сумке должен быть не больше x . Суммарный вес вещей во второй сумке должен быть не больше y . Если возможных ответов несколько, выведите любой из них.

Если распределить вещи по двум сумкам с соблюдением всех ограничений невозможно, выведите «-1».

Примеры

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|------------------------------------|--------------------------|
| 5 10 12 1 4 3 5 10 8 8 5 9 6 | xxxyy |
| 5 10 12 1 4 4 5 10 8 8 8 9 6 | -1 |

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов этой группы, всех тестов всех предыдущих групп, а также всех примеров.

Во всех тестах первой группы $1 \leq n \leq 10$ и $1 \leq x, y, v_i, w_i \leq 100$. За прохождение всех тестов первой группы можно получить 25 баллов.

Во всех тестах второй группы $1 \leq n \leq 100$ и $1 \leq x, y, v_i, w_i \leq 10\,000$. За прохождение всех тестов второй группы можно получить ещё 35 баллов.

В тестах третьей группы $1 \leq n \leq 500$ и $1 \leq x, y, v_i, w_i \leq 250\,000$, то есть дополнительных ограничений нет. За прохождение третьей группы можно получить оставшиеся 40 баллов.

Задача G. Покупки и бонусы

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В интернет-магазине Aweson действуют следующие правила оплаты покупок. Все покупки оплачиваются талерами — внутренней валютой Aweson. Когда новый покупатель регистрируется в системе, для него заводится бонусный счёт, на котором изначально находится 0 талеров. В момент каждой покупки у покупателя есть выбор: либо использовать бонусы, либо копить их. При использовании бонусов можно оплатить любую часть покупки талерами с бонусного счёта, а оставшуюся часть, если что-то осталось, оплатить обычными талерами. При накоплении же пользоваться бонусным счётом для оплаты нельзя, зато на него добавляется ровно 1% от стоимости покупки в талерах.

Михаил составил план своих покупок на Aweson. Теперь он хочет узнать, какой выбор нужно делать при каждой покупке и как их оплачивать, чтобы в итоге потратить как можно меньше обычных талеров.

Формат входных данных

В первой строке записано целое число n — количество покупок ($1 \leq n \leq 100$). Во второй строке заданы n целых чисел p_1, p_2, \dots, p_n через пробел — стоимости первой, второй, ..., n -й покупок в талерах ($0 < p_i \leq 10^5$, все p_i делятся нацело на 100).

Формат выходных данных

В первой строке выведите одно число s — минимальное суммарное количество обычных талеров, с помощью которого Михаил может сделать все свои покупки в заданном порядке.

В следующих n строках выведите стратегию покупок, позволяющую потратить ровно s обычных талеров. Каждая из этих строк должна содержать два целых числа c_k и b_k , разделённых знаком «плюс» и отделённых от него пробелами: сколько обычных талеров и сколько талеров с бонусного счёта Михаилу следует потратить на k -ю покупку. Разумеется, эти числа должны быть неотрицательными, а их сумма должна быть равна стоимости k -й покупки. Если $b_k = 0$, то k -я покупка совершается с накоплением бонусов, а если $b_k > 0$ — с использованием бонусов. Конечно, ни в какой момент Михаил не может потратить бонусов больше, чем их есть, а сумма всех чисел c_k должна быть в точности равна s .

Количество талеров на бонусном счёте после всех n покупок не имеет значения. Если стратегий с минимальным s несколько, выведите любую из них.

Примеры

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|-------------------------|--|
| 2 100 100 | 199 100 + 0 99 + 1 |
| 3 100 10000 100 | 10100 100 + 0 10000 + 0 0 + 100 |

Пояснения к примерам

В первом примере Михаил получает с первой покупки один бонусный талер. Следует потратить его для частичной оплаты второй покупки.

Во втором примере Михаил также получает с первой покупки один бонусный талер. Однако тратить его на второй покупке невыгодно. Вместо этого Михаил может получить ещё 100 талеров со второй покупки. Тогда третью покупку удастся полностью оплатить бонусными талерами, и на бонусном счёту останется ещё один талер.

Задача Н. Команды равной силы (Division 2)

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В этой задаче требуется разделить магов на две команды так, чтобы силы этих команд были равны.

Есть n магов, которых необходимо разделить на две команды для магической игры. Каждому магу сопоставлено целое число: его сила. Сила команды — это сумма сил магов в ней плюс максимум сил магов в ней. Как разделить магов на две команды одинаковой силы?

Формат входных данных

Первая строка ввода содержит одно целое число n : количество магов ($1 \leq n \leq 100$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n : силы магов ($1 \leq a_i \leq 100$).

Формат выходных данных

Если разделить всех n магов на две команды равной силы невозможно, выведите число -1 . В противном случае в первой строке выведите число, равное силе каждой команды, а в следующих четырёх строках выведите описание команд.

Описание одной команды состоит из двух строк. Первая из них содержит одно целое число k — количество магов в этой команде. Вторая строка содержит k целых чисел — номера магов, которые оказались в этой команде. Номера магов в команде можно выводить в любом порядке.

Каждый маг должен быть упомянут ровно один раз ровно в одном из двух описаний команд. Маги нумеруются с единицы в том порядке, в котором их силы заданы во вводе.

Если возможных разделений на команды несколько, выведите одно любое из них.

Примеры

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|-------------------------|--------------------------------|
| 5 1 2 3 4 5 | 12 3 1 3 4 2 2 5 |
| 4 8 1 7 6 | -1 |
| 6 8 6 5 3 8 4 | 24 4 4 6 3 2 2 1 5 |

Пояснения к примерам

В первом примере сила первой команды вычисляется как $(1 + 3 + 4) + 4$, а сила второй команды — как $(2 + 5) + 5$. Получается, что силы обеих команд равны 12.

Во втором примере разделить этих четверых магов на две команды равной силы не представляется возможным.

В третьем примере возможно другое решение — разделение на команду из первого, второго и четвёртого магов (силы $(8 + 6 + 3) + 8 = 25$) и команду из третьего, пятого и шестого магов (силы $(5 + 8 + 4) + 8 = 25$).

Задача I. Команды равной силы

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Решите предыдущую задачу с ограничениями $1 \leq n \leq 500$ и $1 \leq a_i \leq 500$.

Задача J. Сумма кубов

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Ваша задача — найти минимальное целое положительное число, имеющее заданную сумму кубов цифр.

Формат входных данных

Входные данные состоят из одного или нескольких тестовых случаев. Первая строка содержит количество тестовых случаев. Каждый тестовый случай состоит ровно из одного числа n ($1 \leq n \leq 60\,000$) — требуемой суммы кубов.

Формат выходных данных

Вывод для каждого из наборов должен состоять из одной строки, содержащей минимальное положительное число с заданной суммой кубов цифр.

Пример

| <i>стандартный ввод</i> | <i>стандартный вывод</i> |
|-------------------------|--------------------------|
| 2 | 1 |
| 1 | 12 |
| 9 | |