

Задача А. Банкомат

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В некотором государстве в обращении находятся банкноты определённых номиналов. Национальный банк хочет, чтобы банкомат выдавал любую запрошенную сумму при помощи минимального числа банкнот, считая, что запас банкнот каждого номинала неограничен. Помогите Национальному банку решить эту задачу.

Формат входных данных

Первая строка входных данных содержит натуральное число N не превосходящее 100 — количество номиналов банкнот в обращении. Вторая строка входных данных содержит N различных натуральных чисел $x_1, x_2, \dots, x_N$, не превосходящих 10^4 — номиналы банкнот. Третья строка содержит натуральное число S , не превосходящее 10^4 — сумму, которую необходимо выдать.

Формат выходных данных

В первую строку выходного файла выведите минимальное число слагаемых (или -1, если такого представления не существует). Во вторую строку выведите это представление в любом порядке.

Примеры

stdin	stdout
5 1 3 7 12 32 40	3 32 7 1

Задача В. Рюкзак

Имя входного файла: `knapsack.in`
Имя выходного файла: `knapsack.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Найдите максимальный вес золота, который можно унести в рюкзаке вместительностью S , если есть N золотых слитков с заданными весами.

Формат входных данных

В первой строке входного файла записаны два целых числа — S и N ($1 \leq S \leq 10\,000$, $1 \leq N \leq 200$). Далее следует N неотрицательных целых чисел, не превосходящих 100 000 — веса слитков.

Формат выходных данных

Выведите искомый максимальный вес.

Примеры

knapsack.in	knapsack.out
10 3 1 4 8	9
20 4 5 7 12 18	19

Задача С. Рюкзак 0-1: минимум предметов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано N предметов массой $m_1, \dots, m_N$. Ими наполняют рюкзак, который выдерживает вес не более M . Как набрать вес в точности M , используя как можно меньше предметов?

Формат входных данных

В первой строке вводится натуральное число N , не превышающее 100 и натуральное число M , не превышающее 10000.

Во второй строке вводятся N натуральных чисел m_i , не превышающих 100.

Формат выходных данных

Выведите наименьшее необходимое число предметов или 0, если набрать данный вес невозможно.

Примеры

стандартный ввод	стандартный вывод
1 5968 18	0

Задача D. Рюкзак-2

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Найдите максимальную цену слитков золота, которые можно унести в рюкзаке вместительностью S , если есть N золотых слитков с заданными весами и ценами.

Формат входных данных

В первой строке входных данных записаны два числа — S и N ($1 \leq S \leq 10\,000$, $1 \leq N \leq 300$).

В двух следующих строках записано по N неотрицательных целых чисел в каждой — веса и стоимости слитков соответственно. Каждое из этих чисел не превосходит 100 000.

Формат выходных данных

Определите набор предметов максимальной стоимости, помещающийся в данный рюкзак. В первой строке выведите стоимость предметов в набранном наборе, во второй — количество предметов в наборе. В следующей строке выведите через пробел номера этих предметов.

Примеры

stdin	stdout
10 3	123
1 4 8	2
72 7 51	1 3

Задача Е. ЗОРП 2

Имя входного файла: `knapsack-2.in`
Имя выходного файла: `knapsack-2.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Перед вами лежат n котиков. Каждый котик характеризуется своим весом w_i и своей мимимишностью c_i . Вы хотите выбрать некоторое число котиков суммарным весом не более чем S так, чтобы их суммарная мимимишность была максимально возможной.

Формат входных данных

В первой строке содержатся два целых числа n и S — число котиков и максимальный допустимый суммарный вес ($1 \leq n \leq 100$, $1 \leq S \leq 10^9$). Следующие n строк содержат по два целых числа w_i и c_i — вес и мимимишность i -го котика ($1 \leq w_i \leq 10^7$, $0 \leq c_i \leq 10^4$). Гарантируется, что сумма всех c_i не превосходит 10^4 .

Формат выходных данных

В первой строке выведите суммарную мимимишность выбранных котиков. Во вторую строку выведите целое число k — количество выбранных котиков. В третьей строке выведите k чисел — номера выбранных котиков. Если оптимальных ответов несколько, то разрешается вывести любой из них.

Примеры

knapsack-2.in	knapsack-2.out
3 10	11
1 2	2
4 3	3 1
8 9	

Задача F. Получите в сумме ноль

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 146 мегабайт

Даны целые неотрицательные числа $a_1, a_2, \dots, a_n$. Определите, какой знак следует поставить перед каждым из них, чтобы результат сложения данных чисел с учётом выбранных знаков равнялся нулю. Гарантируется, что способ расставить знаки таким образом существует.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 100$). Во второй строке через пробел записаны числа a_i ($0 \leq a_i \leq 500$).

Формат выходных данных

Выведите единственную строку из n символов. i -й символ строки должен соответствовать знаку, который требуется поставить в сумме перед i -м числом, чтобы получить в результате ноль.

Примеры

stdin	stdout
4 2 3 0 1	-+--

Задача G. Подготовка к посвяту

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В ЛКШ приближается посвят — одно из самых ожидаемых событий смены. Небольшим группам детей предстоит пройти по цепочке интересных локаций, справиться со всеми испытаниями подготовленного сценария и добраться до финала. Там каждого участника ждёт очень крутая футболка.

На складе находится n запечатанных коробок с футболками. В коробке с номером i лежит a_i футболок. Всего сформировано t групп, в группе с номером j находится q_j детей.

Организаторы должны выбрать несколько групп, которые пройдут сценарий, и несколько коробок, которые доставят на финальную локацию. Каждый ребёнок из выбранных групп должен получить ровно одну футболку, и все футболки из доставленных коробок должны быть выданы. Поэтому суммарное количество футболок в выбранных коробках должно в точности совпадать с суммарным количеством детей в выбранных группах. Доставлять часть коробки или делить одну группу нельзя.

Определите максимальное количество групп, которые смогут пройти посвят и получить футболки.

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100$) — количество коробок с футболками.

Во второй строке заданы n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^4$) — количество футболок в коробках. Гарантируется, что $\sum a_i \leq 10^4$.

В третьей строке задано целое число t ($1 \leq t \leq 100$) — количество групп.

В четвёртой строке заданы t целых чисел $q_1, q_2, \dots, q_t$ ($1 \leq q_j \leq 10^4$) — количество детей в группах. Гарантируется, что $\sum q_j \leq 10^4$.

Формат выходных данных

Выведите одно целое число — максимальное количество групп, для которых хватит футболок при соблюдении описанных условий.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 3 6 2 3	2
2 10 5 6 1 2 3 5 4 5	5

Замечание

В первом примере можно доставить коробки с 2 и 3 футболками и провести посвят для двух групп, в которых находится соответственно 2 и 3 ребёнка.

Во втором примере можно доставить обе коробки и провести посвят для пяти групп размером 1, 2, 3, 4 и 5 детей. Всего в этих группах 15 детей, а в выбранных коробках — 15 футболок.

Задача Н. Федя и вопросы «Своей игры»

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Как известно, в ЛКШ.2026 Федя ведёт «гитарный клуб». Только что обнаружилось, что для предстоящего квартирника ему не хватает одного сета. Сет в «гитарном клубе» — это набор ровно из пяти песен, относящихся к одному жанру, причём каждая следующая песня должна быть строго круче предыдущей. К счастью, у Феди уже есть набор из N готовых песен в жанре «Бардовские хиты Берляндии». Для каждой из песен известна её крутость (натуральное число). Посчитайте, сколько у Феди способов собрать из песен набора один сет. Два способа считаются различными, если хотя бы одна песня, присутствующая в одном из них, не присутствует во втором.

Формат входных данных

В первой строке файла находится одно число N — количество песен в наборе Феди ($5 \leq N \leq 300$). В следующей строке N чисел, отсортированных по неубыванию — крутости песен из набора. Каждая крутость — натуральное число от 1 до 100, крутости некоторых песен могут совпадать.

Формат выходных данных

Выведите одно число — количество способов составить сет для гитарного клуба.

Примеры

<code>stdin</code>	<code>stdout</code>
7 1 1 2 3 4 5 5	4
7 1 2 3 4 5 5 6	11
5 1 1 2 3 4	0

Задача I. Главное правило личных олимпиад

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

Напомним главное правило написания личных олимпиад: по каждой задаче нужно набрать баллы! Нельзя уйти с конкурса с нулем по задаче.

Промоделируем тур олимпиады. Пусть на туре предложено n задач, i -я задача состоит из k_i подзадач, j -я подзадача i -й задачи приносит $c_{i,j}$ баллов. Зависимостей между подзадачами нет, поэтому можно в каждой задаче выбрать любое множество подзадач и его решить. При этом нельзя выбрать пустое множество, ведь тогда по задаче будет 0 баллов, а это противоречит главному правилу написания личных олимпиад.

Проверьте, можно ли, придерживаясь главного правила личных олимпиад, набрать на туре ровно s баллов.

Формат входных данных

Первая строка содержит два целых числа n, s ($1 \leq n \leq 100\,000$, $1 \leq s \leq 100\,000$) — количество задач в конкурсе и необходимую сумму баллов, соответственно. Далее следуют описания задач. Описание каждой задачи состоит из двух строк.

Первая строка описания i -й задачи содержит одно целое число k_i ($1 \leq k_i \leq 100\,000$) — количество подзадач в i -й задаче.

Вторая строка описания i -й задачи содержит k_i целых чисел $c_{i,1}, c_{i,2}, \dots, c_{i,k_i}$ ($1 \leq c_{i,j} \leq 100\,000$) — баллы за подзадачи.

Гарантируется, что сумма $k_1 + k_2 + \dots + k_n$ по всем задачам не превосходит 100 000.

Гарантируется, что произведение $(k_1 + k_2 + \dots + k_n) \cdot s$ не превосходит 10^7 .

Формат выходных данных

Если решения не существует, выведите «No».

В противном случае в первой строке выведите «Yes». Далее необходимо вывести описание решенных подзадач для каждой задачи.

Описание i -й задачи начинается с целого числа m_i ($1 \leq m_i \leq k_i$) — количества решенных подзадач i -й задачи. Далее следуют m_i различных целых чисел $p_{i,1}, p_{i,2}, \dots, p_{i,m_i}$ ($1 \leq p_{i,j} \leq k_i$) — номера решенных подзадач в i -й задаче.

Если существует несколько подходящих способов набрать s баллов, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
2 4 1 2 2 3 1	No
2 4 1 2 2 2 1	Yes 1 1 1 1