

Задача А. ЗОРП 1

Имя входного файла: `knapsack-1.in`
Имя выходного файла: `knapsack-1.out`
Ограничение по времени: 2 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 ($1 \leq n \leq 100$, $1 \leq S \leq 10^4$). Следующие n строк содержат по два целых числа w_i и c_i — вес и мимимишность i -го котика ($1 \leq w_i \leq 10^4$, $0 \leq c_i \leq 10^7$).

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

Задача В. 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 С. 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 D. ЗОРП 2

Имя входного файла: `knapsack-2.in`
Имя выходного файла: `knapsack-2.out`
Ограничение по времени: 2 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 Е. 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

Задача F. Гирьки: три кучки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан набор гирек массой $m_1, m_2, \dots, m_n$. Можно ли их разбить на три кучки равной массы?

Формат входных данных

Первая строка входных данных содержит натуральное число $N \leq 67$. На следующей строке содержится N натуральных чисел m_i , не превышающих 67.

Формат выходных данных

Программа должна вывести номера гирек для каждого из набора в три строки или -1, если решения не существует.

Примеры

стандартный ввод	стандартный вывод
5 4 2 3 1 5	3 2 5 4 1
2 42 52	-1

Задача G. Распродажа

Имя входного файла: `sale.in`
 Имя выходного файла: `sale.out`
 Ограничение по времени: 3 секунды
 Ограничение по памяти: 256 мегабайт

В супермаркете «На троечку» часто происходят распродажи товаров, срок годности которых подходит к концу. Каждый товар привозят в магазин в определенное время, а через некоторое его вывозят из магазина, в связи с окончанием срока годности. Более формально, каждый товар имеет стоимость c_i , время его завоза в магазин a_i и время его вывоза из магазина b_i .

У Иннокентия есть хитрый план похода в магазин. Даже несколько. Каждый план похода в магазин выглядит так: Иннокентий выбирает какое-то время, когда он появится в магазине m_j , время s_j , которое он проведет в магазине среди огромных стеллажей товаров, и сумму денег k_j , которую он рассчитывает потратить. Для каждого плана он хочет узнать, сможет ли он осуществить его, т. е. верно ли, что он сможет во время своего пребывания в магазине купить несколько товаров суммарной стоимостью **ровно** k_j , при этом все выбранные товары должны быть в магазине на протяжении всего пребывания Иннокентия в магазине.

Помогите Иннокентию определить, какие из его планов можно выполнить.

Формат входных данных

В первой строке входных данных содержится число N — общее количество товаров в магазине ($1 \leq N \leq 800$). Далее содержатся описания товаров, каждый товар описывается тремя целыми числами c_i, a_i, b_i , обозначающими стоимость товара, время его завоза и время его вывоза из магазина ($1 \leq c_i \leq 1\,000, 1 \leq a_i < b_i \leq 10^9$).

Далее содержится число M — количество планов Иннокентия ($1 \leq M \leq 800\,000$). Каждый план описывается тремя целыми числами m_j, k_j, s_j , обозначающими время прихода Иннокентия в магазин, сумму денег, которую он готов потратить в этом плане и длительность его пребывания в магазине ($1 \leq m_j \leq 10^9, 1 \leq k_j \leq 100\,000, 0 \leq s_j \leq 10^9$).

Помните, что это только планы, т. е. ситуация в магазине не меняется вне зависимости от того, может ли Иннокентий осуществить план или нет.

Формат выходных данных

Для каждого плана в отдельной строке выведите «YES», если Иннокентий может его осуществить, и «NO» в противном случае.

Примеры

sale.in	sale.out
5	YES
6 2 7	NO
5 4 9	YES
1 2 4	YES
2 5 8	NO
1 3 9	
5	
2 7 1	
2 7 2	
3 2 0	
5 7 2	
4 1 5	

Задача Н. Делим печенье

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

Василисе и Михаилу подарили n упаковок печенья. Вес i -й упаковки составляет a_i грамм. Они договорились распределить упаковки по следующим правилам:

1. Упаковки рассматриваются последовательно, начиная с первой.
2. На каждом шаге i (от 1 до n) i -ю упаковку забирает тот, у кого на данный момент меньше суммарного веса печенья.
3. Если суммы равны, упаковку забирает Василиса.

Они заметили, что итоговый вес печенья у каждого зависит от порядка упаковок.

Можно ли переупорядочить упаковки так, чтобы после распределения суммы весов у Василисы и Михаила были равны?

Формат входных данных

В первой строке входных данных дано целое число n ($1 \leq n \leq 100$) — количество упаковок.

Во второй строке входных данных дано n целых чисел $a_1, \dots, a_n$ ($1 \leq a_i \leq 100$) — веса упаковок с печеньем.

Формат выходных данных

Если невозможно найти требуемый порядок, выведите -1 .

В противном случае выведите n целых чисел через пробел, задающих корректную перестановку значений a_i , при которой упаковки с печеньем будут разделены поровну между Василисой и Михаилом.

Примеры

стандартный ввод	стандартный вывод
4 2 2 3 1	2 1 3 2
5 1 2 3 4 5	-1