

Задача А. Рюкзак

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

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 числа — S и N ($1 \leq S \leq 10\,000$, $1 \leq N \leq 300$). Далее следует N неотрицательных целых чисел, не превосходящих $100\,000$ — веса слитков.

Формат выходных данных

Выведите искомый максимальный вес.

Пример

knapsack.in	knapsack.out
10 3 5 7 4	9

Задача В. Покупка билетов

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

За билетами на премьеру нового мюзикла выстроилась очередь из N человек, каждый из которых хочет купить 1 билет. На всю очередь работала только одна касса, поэтому продажа билетов шла очень медленно, приводя «постояльцев» очереди в отчаяние. Самые сообразительные быстро заметили, что, как правило, несколько билетов в одни руки кассир продаёт быстрее, чем когда эти же билеты продаются по одному. Поэтому они предложили нескольким подряд стоящим людям отдавать деньги первому из них, чтобы он купил билеты на всех.

Однако для борьбы со спекулянтами кассир продавала не более 3-х билетов в одни руки, поэтому договориться таким образом между собой могли лишь 2 или 3 подряд стоящих человека.

Известно, что на продажу i -му человеку из очереди одного билета кассир тратит A_i секунд, на продажу двух билетов — B_i секунд, трех билетов — C_i секунд. Напишите программу, которая подсчитает минимальное время, за которое могли быть обслужены все покупатели.

Обратите внимание, что билеты на группу объединившихся людей всегда покупает первый из них. Также никто в целях ускорения не покупает лишних билетов (то есть билетов, которые никому не нужны).

Формат входных данных

Во входном файле записано сначала число N — количество покупателей в очереди ($1 \leq N \leq 5000$). Далее идет N троек натуральных чисел A_i , B_i , C_i . Каждое из этих чисел не превышает 3600. Люди в очереди нумеруются начиная от кассы.

Формат выходных данных

В выходной файл выведите одно число — минимальное время в секундах, за которое могли быть обслужены все покупатели.

Пример

tickets.in	tickets.out
5 5 10 15 2 10 15 5 5 5 20 20 1 20 1 1	12

Задача С. ОБЖ

Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Начинается новый учебный год, а с ним и уроки ОБЖ.

В начале урока был отдан приказ построиться по росту.

Оказалось, что 11-классники до сих пор не научились этого делать — они выстроились в линию, но по алфавиту, а не по росту!

Учитель ОБЖ в ярости. Он хочет выгнать часть людей так, чтобы роста оставшихся людей образовывали неубывающую последовательность. (Преподаватель почему-то решил, что строй должен начинаться с низких и заканчиваться высокими.)

Однако, учитель не так прост, как кажется на первый взгляд: он хочет выгонять не кого попало, а тех, кто больше всех выделяется. Формально, выгоняя на каждом шаге ровно одного школьника, он может выгнать только удовлетворяющего одному из условий:

- Школьник строго *ниже* и своего левого ближайшего соседа, и своего правого ближайшего соседа.
- Школьник строго *выше* и своего левого ближайшего соседа, и своего правого ближайшего соседа.

В частности, первого и последнего в строю школьников выгнать нельзя.

Есть и пожелание от директора: количество людей, которых выгонит учитель, должно быть минимальным.

Сможет ли преподаватель добиться своего? Если да, помогите ему выгнать нужных учеников.

Формат входных данных

В первой строке целое n ($2 \leq n \leq 200$) — количество учеников. Во второй строке n целых чисел a_i ($0 < a_i \leq 10^3$) — роста учеников в том порядке, в котором они выстроились изначально.

Формат выходных данных

Если достичь желаемого невозможно, выведите «-1».

В ином случае первая строка должна содержать целое m — минимальное количество учеников, которых придётся выгнать. Каждая из следующих m строк должна содержать одно число b_i — номер ученика, которого надо выгнать на i -ом шаге. Ученики нумеруются в порядке, в котором они выстроились изначально.

Если преподаватель может действовать по-разному, выгнав одно и то же минимальное количество 11-классников, нужно вывести любой способ.

Пример

safe.in	safe.out
5 3 2 4 8 5	2 2 4
5 4 5 5 5 6	0
6 1 1 3 3 2 2	-1
6 400 300 310 300 310 500	-1