

Задача А. Наибольшая неубывающая подпоследовательность

Имя входного файла: `lis.in`
Имя выходного файла: `lis.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Для заданной числовой последовательности a_1, a_2, \dots, a_n требуется найти длину максимальной неубывающей подпоследовательности.

Для неубывающей подпоследовательности $a_{k_1}, a_{k_2}, \dots, a_{k_t}$ ($k_1 < k_2 < \dots < k_t$) верно, что $a_{k_i} \leq a_{k_j}$ при $1 \leq i < j \leq t$. Подпоследовательность из нуля элементов полагается неубывающей по определению.

Формат входного файла

В первой строке входного файла записано одно натуральное число N ($1 \leq N \leq 1000$) — количество чисел в исходной последовательности. Далее следует N натуральных чисел, не превосходящих 10^9 — сама последовательность.

Формат выходного файла

Вывести единственное число, равное искомому количеству.

Пример

<code>lis.in</code>	<code>lis.out</code>
4 3 6 5 12	3

Задача В. Наибольшая последовательнократная подпоследовательность

Имя входного файла: `sequence.in`
Имя выходного файла: `sequence.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Для заданной числовой последовательности a_1, a_2, \dots, a_n требуется найти максимальную последовательнократную подпоследовательность.

Для последовательнократной подпоследовательности $a_{k_1}, a_{k_2}, \dots, a_{k_t}$ ($k_1 < k_2 < \dots < k_t$) верно, что $a_{k_i} | a_{k_j}$ при $1 \leq i < j \leq t$ (утверждение « $a|b$ » эквивалентно « b кратно a »). Подпоследовательность из одного элемента полагается последовательнократной по определению.

Формат входного файла

В первой строке входного файла записано одно натуральное число N ($1 \leq N \leq 1000$) — количество чисел в исходной последовательности. Далее следует N натуральных чисел, не превосходящих 10^9 — сама последовательность.

Формат выходного файла

Выведите последовательнократную подпоследовательность максимальной длины. Если решений несколько — выведите любое.

Пример

<code>sequence.in</code>	<code>sequence.out</code>
4 3 6 5 12	3 6 12

Задача С. Наибольшая общая подпоследовательность

Имя входного файла: `lcs.in`
Имя выходного файла: `lcs.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычеркиванием некоторых элементов).

Формат входного файла

В первой строке входного файла записано число N — длина первой последовательности ($1 \leq N \leq 1000$).

Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 1000$).

В четвертой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

Формат выходного файла

В выходной файл требуется вывести единственное целое число: длину наибольшей общей подпоследовательности, или число 0, если такой не существует.

Пример

lcs.in	lcs.out
3	2
1 2 3	
4	
2 1 3 5	

Задача D. Максимальный подпалиндром

Имя входного файла: palyndrom.in
Имя выходного файла: palyndrom.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Из данной строки удалите наименьшее количество символов так, чтобы получился палиндром (строка, одинаково читающаяся как справа налево, так и слева направо).

Формат входного файла

Во входном файле записана непустая строка, длиной не более 100 символов. Строка состоит только из заглавных латинских букв.

Формат выходного файла

В выходной файл требуется вывести строку-палиндром максимальной длины, которую можно получить из исходной вычеркиванием нескольких букв. При наличии нескольких решений необходимо вывести одно (любое) из них.

Пример

palyndrom.in	palyndrom.out
ASDDFSA	ASDDSA

Задача E. Кино

Имя входного файла: cinema.in
Имя выходного файла: cinema.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Трое друзей летом посмотрели несколько фильмов. Для каждого школьника известно, какие фильмы и в каком порядке он посмотрел (естественно, если фильм кому-то из ребят сильно понравился, он мог его пересмотреть несколько раз). Так как друзья любят смотреть фильмы вместе, втроем они посмотрели максимальное возможное количество. Сколько же раз они встречались вместе?

Формат входного файла

В первой строке входного файла заданы три числа: N , M и K — количества фильмов, просмотренных каждым из друзей ($1 \leq N, M, K \leq 300$). В следующих трех строках выписаны номера фильмов, просмотренных ими. У друзей огромные планы на будущее, поэтому фильмы нумеруются числами между 1 и 10^9 .

Формат выходного файла

В первой строке выведите единственное число — максимальное кол-во просмотренных фильмов.

В следующей строке выведите через пробел номера просмотренных фильмов.

Пример

cinema.in	cinema.out
3 3 4	2
1 2 3	1 3
1 3 10	
3 1 10 3	

Задача F. Покупка билетов

Имя входного файла: tickets.in
Имя выходного файла: tickets.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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 входного файла

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 выходного файла

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

Пример

<code>tickets.in</code>	<code>tickets.out</code>
5	12
5 10 15	
2 10 15	
5 5 5	
20 20 1	
20 1 1	