

Задача А. 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 длину максимальной последовательнократной подпоследовательности.

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 », то есть каждый элемент выбранной подпоследовательности должен быть делителем всех последующих элементов подпоследовательности). Подпоследовательность из одного элемента полагается последовательнократной по определению.

Формат входных данных

В первой строке входных данных записано одно натуральное число N ($1 \leq N \leq 1000$) — количество чисел в исходной последовательности. Во второй строке записано N натуральных чисел, не превосходящих $2 \cdot 10^9$ — сама последовательность.

Формат выходных данных

Программа должна вывести единственное целое число — длину наибольшей последовательнократной подпоследовательности.

Примеры

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

Задача В. Наибольшая общая подпоследовательность

Имя входного файла: `lcs.in`
Имя выходного файла: `lcs.out`
Ограничение по времени: 3 секунды
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 входных данных

В первой строке входных данных записано целое число N — длина первой последовательности ($1 \leq N \leq 1000$). Во второй строке записаны N чисел — члены первой последовательности. В третьей строке записано целое число M — длина второй последовательности ($1 \leq M \leq 1000$). В четвертой строке записаны M чисел — члены второй последовательности. Члены последовательностей — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

Программа должна вывести единственное целое число — длину наибольшей общей подпоследовательности, или число 0, если такой не существует.

Примеры

<code>lcs.in</code>	<code>lcs.out</code>
3 1 2 3 4 2 1 3 5	2

Задача С. 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 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.

Примеры

<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	

Задача D. 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

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

Задача Е. Рюкзак-2

Имя входного файла: `knapsack2.in`
Имя выходного файла: `knapsack2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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

<code>knapsack2.in</code>	<code>knapsack2.out</code>
10 3	123
1 4 8	2
72 7 51	1 3

Задача F. Свёртка

Имя входного файла: `folding.in`
Имя выходного файла: `folding.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Петя хочет сократить запись последовательности, состоящей из заглавных латинских букв. Для этого он может свернуть её повторяющиеся подпоследовательности. Например, последовательность `AAAAAAAAAABABABCCD` может быть записана как `10(A)2(BA)B2(C)D`.

Формальное определение свёрнутой последовательности и соответствующей ей операции развёртки даётся следующим образом:

- Последовательность, которая содержит единственный символ от 'A' до 'Z', представляет из себя свёрнутую последовательность. При развёртке такой последовательности получается она сама.
- Если S и Q — свёрнутые последовательности, то SQ — также свёрнутая последовательность. Если при развёртке строки S получается строка S' , а при развёртке Q получается Q' , то при развёртке SQ получается строка $S'Q'$.
- Если S — свёрнутая последовательность, то $X(S)$ — также свёрнутая последовательность, где X — это десятичное представление целого числа, большего единицы. Если при развёртке строки S получается строка S' , то при развёртке $X(S)$ получается строка S' , повторённая X раз.

Петя хочет свернуть заданную последовательность таким образом, чтобы результат содержал наименьшее число символов.

Формат входных данных

Входной файл содержит непустую строку, состоящую из заглавных латинских букв. Длина строки не превышает 100 символов.

Формат выходных данных

В выходной файл выведите одну строку, содержащую наименьшую последовательность, развёртка которой даст строку, заданную во входном файле.

Если ответов несколько, выведите любой из них.

Примеры

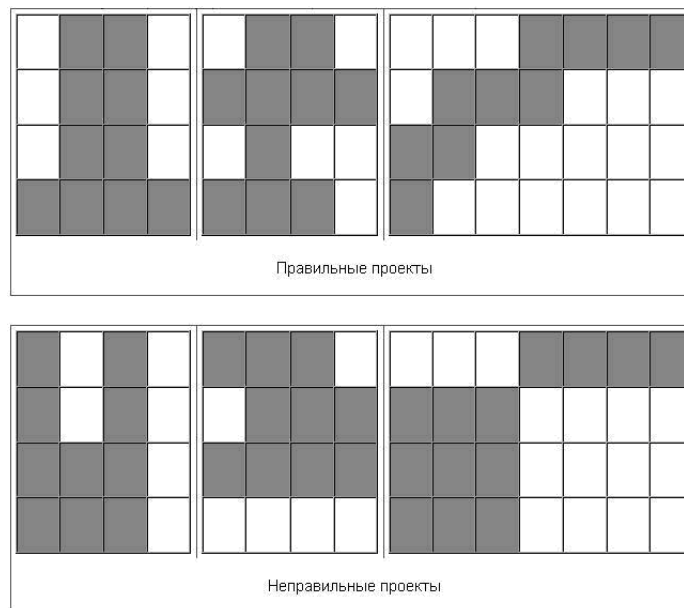
<code>folding.in</code>	<code>folding.out</code>
<code>AAAAAAAAAABABABCCD</code>	<code>9(A)3(AB)CCD</code>
<code>NEERCYESYESYESNEERCYESYESYES</code>	<code>2(NEERC3(YES))</code>

Задача G. Авангардная архитектура

Имя входного файла: `avant-garde.in`
Имя выходного файла: `avant-garde.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Один из столичных девелоперов решил построить жилой дом по проекту известного авангардного архитектора. Жилой дом будет состоять из квартир-кубиков и иметь причудливую форму. Есть два ограничения, одно из которых наложено архитектором, а второе — законами физики.

Архитектор хочет, чтобы каждый этаж представлял собой связанную последовательность кубиков (разделённые этажи — это мода 90-х). В то же время необходимо, чтобы хотя бы под одним из кубиков этажа находился кубик предыдущего этажа. Первый этаж должен опираться о землю.



Кроме законов физики архитектора ограничивает также необходимость всё это творчество продать. Поскольку покупатели неохотно покупают недвижимость, необходимо привлечь их хоть чем-нибудь, в частности, видом из окна. Специалисты компании-девелопера составили таблицу, в которой для каждого возможного расположения квартиры указана привлекательность вида из окна для этого расположения. Необходимо максимизировать суммарную привлекательность видов из окна.

В приведённом примере показаны привлекательности видов из окна и наилучшее здание из 10 кубиков в данном случае.

9	3	6	4	8	1	3
2	9	2	5	3	2	6
1	1	8	4	6	5	4
1	9	6	5	3	4	5
6	2	5	6	7	1	2
2	6	7	5	6	4	3

По известному количеству кубиков и таблице привлекательности видов из окна вам необходимо выбрать лучший проект (с максимальной суммарной привлекательностью), удовлетворяющий условиям архитектора и законам физики.

Формат входных данных

В первой строке входного файла указаны натуральные числа N , H и W ($1 \leq H \leq 30$, $1 \leq W \leq 30$, $1 \leq N \leq HW$) — количество имеющихся кубиков, максимальная высота и максимальная ширина здания. Следующие H строк содержат по W натуральных чисел, задающих привлекательность соответствующего расположения квартиры. Привлекательность измеряется в пределах от 1 до 100 000 включительно.

Формат выходных данных

Выведите одно число — наибольшую суммарную привлекательность.

Примеры

avant-garde.in	avant-garde.out
10 6 7 9 3 6 4 8 1 3 2 9 2 5 3 2 6 1 1 8 4 6 5 4 1 9 6 5 3 4 5 6 2 5 6 7 1 2 2 6 7 5 6 4 3	65

Задача Н. Гладиолус

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

Числовая последовательность задана рекуррентной формулой: $a_{i+1} = (ka_i + b) \bmod m$. Найдите её наибольшую возрастающую подпоследовательность.

Формат входных данных

Программа получает на вход пять целых чисел: длину последовательности n ($1 \leq n \leq 10^5$), начальный элемент последовательности a_1 , параметры k , b , m для вычисления последующих членов последовательности ($1 \leq m \leq 10^4$, $0 \leq k < m$, $0 \leq b < m$, $0 \leq a_1 < m$).

Формат выходных данных

Требуется вывести наибольшую возрастающую подпоследовательность данной последовательности, разделяя числа пробелами. Если таких последовательностей несколько, необходимо вывести одну (любую) из них.

Примеры

<code>gladiolus.in</code>	<code>gladiolus.out</code>
5 41 2 1 100	41 67 71

Замечание

В данном примере последовательность состоит из 5 элементов: $a_1 = 41$, $a_{i+1} = (2a_i + 1) \bmod 100$, то есть последовательность имеет вид 41, 83, 67, 35, 71.