

Задача А. Наибольшая возрастающая подпоследовательность

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

Дана последовательность, требуется найти длину её наибольшей строго возрастающей подпоследовательности. Подпоследовательность — это часть последовательности, получающаяся из нее вычеркиванием каких-то элементов.

Формат входных данных

В первой строке входного файла задано целое число N — длина последовательности ($1 \leq N \leq 1000$). Во второй строке задаётся сама последовательность (разделитель — пробел). Элементы последовательности — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

Требуется вывести длину наибольшей возрастающей подпоследовательности.

Примеры

<code>lis.in</code>	<code>lis.out</code>
6 3 29 5 5 28 6	3

Задача В. Гладиолус

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

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.

Задача С. 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.2013.Июль «Свою игру» проводит Федя. Только что обнаружилось, что для предстоящего турнира ему не хватает одной темы. Тема в «Своей игре» — это набор ровно из пяти вопросов, лежащих в одной области, причем каждый следующий вопрос строго сложнее предыдущего. К счастью, у Феде уже есть пакет из N готовых вопросов из области «Исторические личности Берляндии». Для каждого из вопросов известна его сложность (натуральное число). Посчитайте, сколько у Феде способов собрать из вопросов пакета одну тему. Два способа считаются различными, если хотя бы один вопрос, присутствующий в одном из них, не присутствует во втором.

Формат входных данных

В первой строке файла находится одно число N — количество вопросов в наборе Феде ($5 \leq N \leq 300$). В следующей строке N чисел, отсортированных по неубыванию — сложности вопросов из пакета. Каждая сложность — натуральное число от 1 до 100, сложности некоторых вопросов могут совпадать.

Формат выходных данных

Выведите одно число — количество способов сделать тему для «Своей игры».

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 D. 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 определенных 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

<code>stdin</code>	<code>stdout</code>
5 1 3 7 12 32 40	3 32 7 1

Задача Е. Сумма кубов

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

Дано натуральное число N . Необходимо представить его в виде суммы точных кубов натуральных чисел, содержащей наименьшее число слагаемых. Программа должна вывести это число слагаемых.

Формат входных данных

Программа получает на вход натуральное число N , не превосходящее $2^{15} - 1$.

Формат выходных данных

Программа должна вывести единственное натуральное число.

Примеры

<code>cubes.in</code>	<code>cubes.out</code>
33	5
1	1

Задача F. Рюкзак

Имя входного файла: `knapsack.in`
Имя выходного файла: `knapsack.out`
Ограничение по времени: 2 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 записано одно число — S ($1 \leq S \leq 10\,000$).

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

Формат выходных данных

Выведите искомый максимальный вес.

Примеры

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

Задача G. 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

<code>stdin</code>	<code>stdout</code>
10 3	123
1 4 8	2
72 7 51	1 3

Задача N. Наибольшая общая подпоследовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 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 входного файла через пробел записаны N членов первой последовательности ($1 \leq N \leq 1000$) — целых чисел, не превосходящих 10 000 по модулю. Во второй строке через пробел записаны M членов второй последовательности ($1 \leq M \leq 1000$) — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

В первую строку выходного файла требуется вывести единственное целое число: длину наибольшей общей подпоследовательности или число 0, если такой не существует. Во вторую строку выходного файла требуется вывести самую наибольшую общую подпоследовательность, через пробел (если подпоследовательностей несколько, выведите любую).

Примеры

стандартный ввод	стандартный вывод
1 2 3	2
2 1 3 5	2 3

Задача I. Наибольшая последовательнократная подпоследовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
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 »). Подпоследовательность из одного элемента полагается последовательнократной по определению.

Формат входных данных

В первой строке входного файла записаны N натуральных чисел ($1 \leq N \leq 1000$), не превосходящих $2 \cdot 10^9$ — последовательность.

Формат выходных данных

Вывести единственное число, равное длине максимальной последовательнократной подпоследовательности.

Примеры

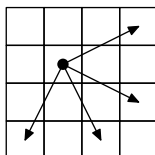
стандартный ввод	стандартный вывод
3 6 5 12	3

Задача J. Ход конём - 2

Имя входного файла: `knight2.in`
Имя выходного файла: `knight2.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана прямоугольная доска $N \times M$ (N строк и M столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

Формат входных данных

Входной файл содержит два натуральных числа N и M ($1 \leq N, M \leq 50$). Гарантируется, что ответ на задачу не превышает 1000000000.

Формат выходных данных

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

Примеры

<code>knight2.in</code>	<code>knight2.out</code>
4 4	2
2 3	1

Задача К. Собеседование в «Отбой»

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

Вы проходите собеседование в легендарную команду «Отбой». Вы успешно отвечали на все вопросы, и теперь от главного приза — членства в команде «Отбой», дающего доступ к безлимитным запасам кефирчика, вас отделяет последняя задача — посчитать количество чисел на отрезке с l по r , сумма цифр которых кратна числу k . Более формально — вам нужно посчитать количество чисел i ($l \leq i \leq r$), у которых сумма цифр кратна k .

Формат входных данных

В единственной строке входного файла заданы три числа l, r, k ($1 \leq l \leq r \leq 10^{18}, 1 \leq k \leq 1000$).

Формат выходных данных

В единственную строку выходного файла выведите ответ на задачу.

Примеры

<code>coolnumbers.in</code>	<code>coolnumbers.out</code>
1 10 1	10
1 10 2	4

Задача L. Транзисторы над Пекином

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Всемирно известный профессор В.В. Адимов продолжает свои разнообразные исследование устойчивости транзисторов. Теперь в голову ему пришла следующая задача: в доме N этажей, профессор хочет выяснить номер максимального этажа, падение с которого оставляет транзистор целым. Поскольку профессор исследует сферические транзисторы в вакууме, то можете считать что разбившись при падении с этажа f транзистор обязательно разобьется при падении с этажа $f + 1$. Дополнительно поставлено условие, что разрешено проведение не более чем K испытаний.

Эта задача была поручена именно вам, как самому успешному аспиранту профессора Адимова. Поскольку транзисторы нынче в цене, но наука все-таки дороже, то необходимо выяснить, какое минимальное количество транзисторов необходимо закупить, чтобы успешно провести эксперимент даже если вам будет катастрофически не везти.

Формат входных данных

В первой и единственной строке входного файла содержатся два целых положительных числа N и K не превосходящих 2000.

Формат выходных данных

Выведите единственное число - ответ на поставленную задачу. Если для данных N и K возможна ситуация, при которой мы не сможем получить ответ на вопрос даже имея неограниченный запас бесплатных транзисторов выведите -1 .

Примеры

	<code>stdin</code>	<code>stdout</code>
1	4 2	-1
2	4 3	2