Задача A. 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 ($1 \leqslant N \leqslant 1000$) — целых чисел, не превосходящих 10 000 по модулю. Во второй строке через пробел записаны M членов второй последовательности ($1 \leqslant M \leqslant 1000$) — 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 саму наибольшую общую подпоследовательность, через пробел (если подпоследовательностей несколько, выведите любую).

Примеры

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

Задача B. 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 ($1 \leqslant n \leqslant 10^5$), начальный элемент последовательности a_1, параметры k, b, m для вычисления последующих членов последовательности ($1 \leqslant m \leqslant 10^4$, $0 \leqslant k < m$, $0 \leqslant b < m$, $0 \leqslant a_1 < m$).

Формат выходных данных

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

Примеры

gladiolus.in	gladiolus.out
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 C. 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 ($1 \leqslant S \leqslant 10\,000$, $1 \leqslant N \leqslant 300$).

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

stdin	stdout
10 3 1 4 8 72 7 51	123 2 1 3

Задача D. Рюкзак 0-1: минимум предметов

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

Дано N предметов массой $m_1, \ldots, m_N$. Ими наполняют рюкзак, который выдерживает вес не более M. Как набрать вес в точности M, используя как можно меньше предметов?

Формат входных данных

В первой строке вводится натуральное число N, не превышающее 100 и натуральное число M, не превышающее 10000.

Во второе строке вводятся N натуральных чисел m_i, не превышающих 100.

Формат выходных данных

Выведите наименьшее необходимое число предметов или 0, если набрать данный вес невозможно.

Примеры

стандартный ввод	стандартный вывод
1 5968 18	0

Задача E. Кино

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

В первой строке входного файла заданы три числа: N, M и K — количества фильмов, просмотренных каждым из друзей ($1 \leqslant N, M, K \leqslant 100$). 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 выходных данных

В первой строке выведите единственное число — максимальное кол-во просмотренных фильмов. В следующей строке выведите через пробел номера просмотренных фильмов.

Примеры

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

Задача F. Строка

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

Дед Мороз составил важное сообщение, которое нужно передать другим Дедам Морозам.На всякий случай Дед Мороз записал свое сообщение два раза.

Потом он ненадолго отвлекся, а в это время один из эльфов прибежал в дом и решил пошалить. Он взял и приписал к сообщениям спереди и сзади какую-то абракадабру.

Теперь Деду Морозу надо найти наибольшую подстроку, которая могла бы быть его сообщением. Помогите ему.

Например, пусть сообщение Деда Мороза имело вид ‘`xxz`’. После этого эльф приписал к первому сообщению спереди букву `a`, сзади — ничего; ко второму — спереди ничего, сзади `b`. В итоге получились строки ‘`axxz`’ и ‘`xxzb`’.

Формат входных данных

Входной файл содержит две строки, состоящие из строчных латинских букв, длиной не более 5000. Гарантируется, что обе строки во входном файле непустые.

Формат выходных данных

Выведите строку — предполагаемое сообщение.

Примеры

lcstr.in	lcstr.out
caa aab	aa

Задача G. ЗОРП 2

Имя входного файла:	`knapsack-2.in`
Имя выходного файла:	`knapsack-2.out`
Ограничение по времени:	4 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 ($1 \leqslant n \leqslant 100$, $1 \leqslant S \leqslant 10^9$). Следующие n строк содержат по два целых числа w_i и c_i — вес и мимишность i-го котика ($1 \leqslant w_i \leqslant 10^7$, $0 \leqslant c_i \leqslant 10^4$). Гарантируется, что сумма всех c_i не превосходит 10^4.

Формат выходных данных

В первой строке выведите суммарную мимишность 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 1 2 4 3 8 9	11 2 3 1

Задача H. Удаление скобок - 2

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

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

Формат входных данных

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

Формат выходных данных

Выведите строку максимальной длины, являющуюся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов. Если возможных ответов несколько, выведите любой из них.

Примеры

stdin	stdout
([)]	[]
{([(]{)})]	([]{})
]{)[	

Задача I. Распил брусьев

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

Вам нужно распилить деревянный брус на несколько кусков в заданных местах. Распилочная компания берет k рублей за распил одного бруска длиной k метров на две части.

Понятно, что различные способы распила приводят к различной суммарной стоимости заказа. Например, рассмотрим брус длиной 10 метров, который нужно распилить на расстоянии 2, 4 и 7 м, считая от одного конца. Это можно сделать несколькими способами. Можно распилить сначала на отметке 2 м, потом 4 и, наконец, 7 м. Это приведет к стоимости $10+8+6=24$, потому что сначала длина бруса, который пилили, была 10 м, затем она стала 8 м, и, наконец, 6 м. А можно распилить иначе: сначала на отметке 4 м, затем 2, затем 7м. Это приведет к стоимости $10+4+6=20$, что лучше.

Определите минимальную стоимость распила бруса на заданные части.

Формат входных данных

Первая строка входных данных содержит целое число L ($2 \leqslant L \leqslant 10^6$) - длину бруса и целое число N ($1 \leqslant N \leqslant 100$) - количество распилов. Во второй строке записано N целых чисел C_i ($0 < C_i < L$) в строго возрастающем порядке - места, в которых нужно сделать распилы.

Формат выходных данных

Выведите одно натуральное число - минимальную стоимость распила.

Примеры

stdin	stdout
10 3 2 4 7	20
100 3 15 50 75	200