Содержание

1 Задача A. Три последовательности

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

Даны три последовательности целых чисел. Ваша задача — найти их наибольшую общую подпоследовательность.

Формат входных данных

Входной файл содержит описание трех последовательностей. Каждая последовательность задается двумя строчками. Первая строка содержит длину последовательности n ($1 \leqslant n \leqslant 100$), а вторая — ее элементы (32-х битные целые числа).

Формат выходных данных

Первая строка выходного файла должна содержать длину максимальной общей подпоследовательности. Саму подпоследовательность необходимо вывести во второй строке. Если таких строк несколько, можно вывести любую из них.

Примеры

threeseq.in	threeseq.out
3 1 2 3 3 2 1 3 3 1 3 5	2 1 3
3 1 2 3 3 4 5 6 3 1 3 5	0

2 Задача B. Удаление скобок — 2

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

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

erase.in	erase.out
`([)]`	`[]`
`{([(]{)})]`	`([]{})`
`]{)[`	

3 Задача C. Беру? Нет, не беру!

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

Саша и Петя играют в весёлую игру. Петя рисует на листе бумаги таблицу $m \times n$ и заполняет её целыми числами. После этого Саша ставит свою фишку на клетку $(1, 1)$ и может каждым своим ходом двигать её вправо или вниз. Кроме того, если Сашина фишка находится на числе большем, чем все числа, которые он брал до этого, Саша может сказать «беру», и тогда Петя записывает на бумажку то число, на котором стоит Сашина фишка (если Саша не брал ранее ни одного числа, то он может сказать «беру» на любом числе). Игра заканчивается, когда Сашина фишка больше не может ходить, и количество очков, которые набрал Саша, зависит от того, сколько чисел он взял. Помогите Саше «взять» как можно больше чисел!

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных (не более 5). Набор входных данных начинается строкой, в которой записаны числа m и n ($1 \leqslant m, n \leqslant 100$). Далее следуют n строк по m чисел в каждой — таблица, которую нарисовал Петя. Все числа не превосходят по модулю 10000.

Файл завершается набором с $m = n = 0$.

Формат выходных данных

Для каждого из наборов входных данных выведите в первой строке максимальное количество чисел M, которые может взять Саша. Во второй строке выведите один из возможных вариантов Сашиных действий — строку из символов `R`, `D` и `T`, где `R` означает ход вправо ($x \leftarrow x+1$), `D` обозначает ход вниз ($y \leftarrow y + 1$) и `T` обозначает взятие числа, на котором в данный момент стоит фишка.

Разделяйте вывод для разных наборов входных данных одной пустой строкой.

Пример

take.in	take.out
2 2 1 1 1 1 2 3 1 2 1 1 1 3 0 0	1 TRD 3 TRTDDT

4 Задача D. Коммивояжёр возвращается!

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

Коммивояжёр возвращается в систему Альфы Центавра! Население системы с нетерпением ждёт его прибытия — каждый хочет приобрести что-нибудь с далёких планет!

Как обычно, коммивояжёр хочет минимизировать транспортные расходы. Он выбирает начальную планету, прилетает туда на межгалактическом корабле, после чего посещает все остальные планеты системы в порядке, минимизирующем суммарную стоимость посещения, и на другом межгалактическом корабле улетает обратно. Естественно, коммивояжёр не хочет летать ни на какую планету дважды.

Найдите оптимальный маршрут для коммивояжёра. Массы больше не могут ждать!

Формат входных данных

В системе Альфы Центавра n планет. Это число записано в первой строке входного файла ($1 \leqslant n \leqslant 19$). Следующие n строк содержат по n чисел каждая: j-ое число на i-ой из этих строк — стоимость перемещения a_{ij} от i-ой планеты до j-ой. Числа в каждой строке разделены пробелами. Числа a_{ii} не несут полезной информации. Все числа во входном файле положительны и не превосходят 10^8.

Формат выходных данных

В первой строке выходного файла выведите минимальную суммарную стоимость посещения всех планет. Во второй строке выведите n чисел через пробел — номера планет системы в порядке их посещения. Если оптимальных маршрутов несколько, можно вывести любой из них.

Пример

salesman.in	salesman.out
3 8 1 6 3 5 7 4 9 2	5 3 1 2

5 Задача E. Отметки на подмножествах

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

Рассмотрим множество $\mathcal{S}$, состоящее из n элементов — натуральных чисел 1, 2, ..., n.

Сперва отметим несколько подмножеств $\mathcal{S}$, а также все подмножества этих подмножеств.

Затем снимем все отметки, если они есть, с нескольких подмножеств $\mathcal{S}$, а также со всех их подмножеств.

Найдите количество отмеченных подмножеств после всех этих операций.

Формат входных данных

В первой строке входного файла заданы через пробел три целых числа n, x и y. Следующие x строк содержат описания подмножеств, отмеченных на первом шаге, по одному на строке; также были отмечены все подмножества этих подмножеств. Наконец, последние y строк входного файла содержат описания подмножеств, с которых сняли отметки на втором шаге, по одному на строке; также были сняты отметки со всех их подмножеств. Описание каждого подмножества имеет вид k a_1 a_2 ... a_k, где k — количество элементов данного подмножества ($0 \leqslant k \leqslant n$), а a_i — сами элементы (a_i попарно различны, $1 \leqslant a_i \leqslant n$). Элементы могут быть перечислены в любом порядке.

Формат выходных данных

В первой строке выходного файла выведите одно число — количество отмеченных подмножеств после всех описанных операций.

Ограничения

- $1 \leqslant n \leqslant 10$
- $0 \leqslant x, y \leqslant 1000$

Примеры

marked.in	marked.out
1 1 1 1 1 0	1
2 0 1 2 2 1	0
3 2 1 2 1 2 2 2 3 2 1 3	3

Пояснения к примерам

В первом примере на первом шаге ставится отметка на подмножество {1} и на пустое подмножество, на втором шаге с пустого подмножества снимается отметка.

Во втором примере отметок нет.

В третьем примере на первом шаге отмеченными оказываются следующие шесть подмножеств: {}, {1}, {2}, {3}, {1, 2} и {2, 3}. На втором шаге снимаются отметки с трёх подмножеств {}, {1} и {3}.

6 Задача F. Отметки на подмножествах 2

Имя входного файла:	`marked2.in`
Имя выходного файла:	`marked2.out`
Ограничение по времени:	2 секунд
Ограничение по памяти:	256 мегабайт

Условие такое же, как у задачи marked.

Ограничения

- $1 \leqslant n \leqslant 20$
- $0 \leqslant x, y \leqslant 1000$