

Задача А. 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

<code>stdin</code>	<code>stdout</code>
<code>([])</code>	<code>[]</code>
<code>{([[]{}])}</code>	<code>([]{})</code>
<code>]{}[</code>	

Задача В. Коды, сохраняющие порядок

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

Двоичный код — это код, где каждому символу сопоставляется последовательность из единиц и нулей. Код называется префиксным, если ни одно кодовое слово не является префиксом другого. Код называется сохраняющим порядок, если лексикографический порядок кодовых слов совпадает с алфавитным порядком символов.

Рассмотрим текст над алфавитом, содержащим n символов, в котором a_1 раз встречается первый символ, a_2 раз встречается второй символ, \dots , a_n раз встречается n -й символ. Длина текста после кодирования его префиксным кодом, где первому символу сопоставлена строка длины l_1 , второму — строка длины l_2 , и т. д., будет равна $a_1 \cdot l_1 + a_2 \cdot l_2 + \dots + a_n \cdot l_n$.

Требуется найти сохраняющий порядок префиксный код, минимизирующий длину закодированного текста.

Формат входных данных

Первая строка содержит число n — число символов в алфавите ($2 \leq n \leq 2000$). Следующая строка содержит n целых чисел — сколько раз каждый символ встречается в тексте: a_1, a_2, \dots, a_n . Числа положительные и не превосходят 10^9 .

Формат выходных данных

Выведите n двоичных последовательностей — искомый код.

Примеры

<code>codes.in</code>	<code>codes.out</code>
5	00
1 8 2 3 1	01
	10
	110
	111

Задача С. Сокровища

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

Дочь короля Флатландии собирается выйти за прекрасного принца. Принц хочет подарить принцессе сокровища, но он не уверен какие именно бриллианты из своей коллекции выбрать.

В коллекции принца n бриллиантов, каждый характеризуется весом w_i и стоимостью v_i . Принц хочет подарить наиболее дорогие бриллианты, однако король умен и не примет бриллиантов суммарного веса больше R . С другой стороны, принц будет считать себя жадным всю оставшуюся жизнь, если подарит бриллиантов суммарным весом меньше L .

Помогите принцу выбрать набор бриллиантов наибольшей суммарной стоимости, чтобы суммарный вес был в отрезке $[L, R]$.

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 32$), L и R ($0 \leq L \leq R \leq 10^{18}$). Следующие n строк описывают бриллианты и содержит по два числа — вес и стоимость соответствующего бриллианта ($1 \leq w_i, v_i \leq 10^{15}$).

Формат выходных данных

Первая строка вывода должна содержать k — количество бриллиантов, которые нужно подарить принцессе. Вторая строка должна содержать номера даримых бриллиантов.

Бриллианты нумеруются от 1 до n в порядке появления во входных данных.

Если составить подарок принцессе невозможно, то выведите 0 в первой строке вывода.

Примеры

dowry.in	dowry.out
3 6 8	1
3 10	2
7 3	
8 2	

Задача D. Свёртка

Имя входного файла:	stdin
Имя выходного файла:	stdout
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 АААААААААВАВ АВ ВССD 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

stdin	stdout
АААААААААВАВ АВ ВССD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

Задача Е. Дубы

Имя входного файла:	<code>oaks.in</code>
Имя выходного файла:	<code>oaks.out</code>
Ограничение по времени:	1.5 секунды
Ограничение по памяти:	256 мегабайт

На аллее перед зданием Министерства Обороны в ряд высажены n дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

Формат входных данных

Первая строка входного файла содержит целое число n — количество дубов, растущих на аллее ($2 \leq n \leq 200$). Вторая строка содержит n чисел — высоты дубов, приведённые слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

Формат выходных данных

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число -1 .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число m — минимальное количество дубов, которые необходимо срубить. В следующие m строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до n .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

Примеры

<code>oaks.in</code>	<code>oaks.out</code>
5	2
3 2 4 8 5	2
	4

Задача F. Покраска забора

Имя входного файла:	paint.in
Имя выходного файла:	paint.out
Ограничение по времени:	0.7 секунда
Ограничение по памяти:	256 мегабайт

Мэр города Многоярославца решил построить перед своим домом забор из n деревянных досок и нанять лучшего маляра города для его покраски. Поскольку забор должен стать главной достопримечательностью города, лучший дизайнер города для каждой доски назначил тщательно выбранный цвет, в который она должна быть покрашена.

Для покраски главный маляр решил применить новейшую технологию, специально разработанную им для выполнения этого задания. Покраской забора будет заниматься специальный робот, который за один час может покрасить произвольный отрезок забора (набор соседних досок) в некоторый цвет. Поскольку задание должно быть выполнено как можно быстрее, требуется составить программу для робота, которая позволит достичь требуемой раскраски за минимальное время. Оставить какую-то из досок непокрашенной, естественно, запрещается.

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 300$), где n количество досок в заборе. Вторая строка содержит строку из n символов, описывающую требуемую покраску забора. Цвета обозначаются заглавными латинскими буквами.

Формат выходных данных

В первой строке выходного файла выведите m - наименьшее возможное время покраски забора в часах. Следующие m строк должны содержать программу покраски для робота. Каждая строка должна содержать два числа l_i и r_i , а также заглавную букву латинского алфавита, задающую цвет c_i и означает, что робот должен покрасить участок забора с l_i по r_i - доску в цвет c_i (если длина забора n , должно выполняться $1 \leq l_i \leq r_i \leq n$).

Примеры

paint.in	paint.out
5 ABBCA	3 1 5 A 4 4 C 2 3 B
2 AA	1 1 2 A

Задача G. Серверы

Имя входного файла: `server.in`
Имя выходного файла: `server.out`
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Компьютерная сеть в некотором доме строилась по принципу присоединения нового компьютера к последнему из уже подключенных. Никакие два компьютера, будучи подключенными в сеть, между собой дополнительно никак не связывались. Таким образом, в сеть были объединены последовательно N компьютеров. Соседи обменивались информацией между собой, но в какой-то момент поняли, что им нужны прокси-серверы. Компьютерное сообщество дома решило установить прокси-серверы ровно на K компьютеров. Осталось только решить, какие именно компьютеры выбрать для этой цели. Главным критерием является ежемесячная стоимость обслуживания серверами всех компьютеров.

Для каждого компьютера установлен тариф его обслуживания, выраженный в рублях за метр провода. Стоимость обслуживания одного компьютера каким-то сервером равна тарифу компьютера, умноженному на суммарную длину провода от этого компьютера до сервера, которым он обслуживается.

Ваша задача написать программу, которая выберет такие K компьютеров, чтобы установить на них прокси-серверы, что общие затраты на обслуживание всех компьютеров были бы минимальными.

Формат входных данных

В первой строке входного файла записано два целых числа N и K — количество компьютеров в сети и количество прокси-серверов, которые нужно установить ($1 \leq K \leq N \leq 2000$).

Все компьютеры в сети пронумерованы числами от 1 до N по порядку подключения.

Во второй строке записано одно целое число T_1 — тариф обслуживания первого компьютера.

В следующих $N - 1$ строках записано через пробел по два целых неотрицательных числа L_i , T_i — информация об остальных компьютерах в сети по порядку номеров. L_i — длина провода, соединяющего i -й компьютер с соседним с меньшим номером, T_i — тариф обслуживания данного компьютера ($2 \leq i \leq N$). Все L_i и T_i от 0 до 10^6 .

Формат выходных данных

В первую строку выходного файла необходимо вывести одно целое число — минимальную стоимость обслуживания всех компьютеров всеми серверами. Во второй строке должны быть записаны через пробел K номеров компьютеров, на которые необходимо установить серверы (номера разрешается выводить в любом порядке). При существовании нескольких вариантов размещения разрешается вывести любой.

Примеры

<code>server.in</code>	<code>server.out</code>
3 1 10	19 1
2 2 3 3	
3 2 10	4 1 3
2 2 3 3	

Задача Н. Мафия в городе

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

Об этом еще никто не знает, но многие догадываются — мафия уже в городе. Поговаривают, что в планах главы мафиозного клана захват контроля над всем городом, однако поначалу он решил ограничиться захватом основных линий связи города.

В городе находятся n базовых телефонных станций, некоторые пары которых соединены двусторонними каналами связи. Для удобства, занумеруем базовые станции целыми числами от 1 до n , канал связи в этом случае задается парой чисел (u, v) — номерами станций, которые он соединяет.

Будем говорить, что канал связи (u, v) *контролируется* мафией, если захвачена либо станция u , либо станция v (либо обе).

Глава мафиозного клана хочет контролировать все каналы связи, захватив при этом как можно меньше базовых станций. Ваша задача — помочь службе безопасности телефонной компании, составив возможный план захвата и определив количество таких планов.

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m ($2 \leq n \leq 18$, $0 \leq m$). Каждая из последующих m строк описывает один канал связи и содержит по два целых числа: u и v ($1 \leq u, v \leq n$, $u \neq v$) — номера базовых станций, соединенных этим каналом связи. Любая пара станций соединена не более, чем одним каналом.

Формат выходных данных

В первой строке входного файла выведите два числа: k и c — соответственно, минимальное количество базовых станций, которые необходимо захватить для того, чтобы контролировать все каналы связи, и число способов захватить такое количество станций, так чтобы контролировать все каналы связи.

Во второй строке входного файла выведите k чисел — номера базовых станций, соответствующих одному из способов захвата.

Примеры

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

Замечание

В первом примере существует три способа захватить две станции так, чтобы контролировать все каналы связи: $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$.

Задача I. Максимальное среднее

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

На доске записана последовательность натуральных чисел. За один ход разрешается выбрать два соседних числа в последовательности и заменить их на их среднее арифметическое. Будем делать ходы до тех пор, пока на доске не останется одно число. Какое максимальное число может получиться?

Формат входных данных

В первой строке входного потока задано целое число n — количество чисел в последовательности ($1 \leq n \leq 100$). Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n через пробел — сама последовательность ($1 \leq a_i \leq 10\,000$).

Формат выходных данных

Выведите единственное число — максимальное число, которое может получиться в результате. Ответ считается верным, если он отличается от точного ответа не более чем на 10^{-6} по абсолютной или относительной погрешности.

Пример

стандартный ввод	стандартный вывод
3 1 2 3	2.25

Задача J. Сокращение многоугольника

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

Пусть задан некоторый выпуклый n -угольник. Сокращенным многоугольником назовем невырожденный выпуклый многоугольник, множество вершин которого является подмножеством множества вершин исходного многоугольника. Напомним, что многоугольник называется невырожденным, если его площадь больше нуля.

Ваша задача — найти сокращенный многоугольник наибольшей площади среди тех сокращенных многоугольников, периметр которых не превышает P .

Формат входных данных

Первая строка входного файла содержит количество вершин многоугольника n ($3 \leq n \leq 20$) и целое число P ($0 \leq P \leq 10^4$).

Каждая из последующих n строк содержит по два целых числа — координаты соответствующей вершины многоугольника. Вершины заданы в порядке обхода по часовой стрелке, их координаты не превосходят по модулю 100.

Никакие три последовательные вершины многоугольника не лежат на одной прямой.

Формат выходных данных

В выходной файл выведите наибольшую площадь искомого многоугольника. При проверке ответ будет считаться правильным, если он отличается от правильного не более, чем на 10^{-4} .

Если ни одного удовлетворяющего ограничениям сокращенного многоугольника не существует, выведите в выходной файл число 0.

Примеры

<code>reduce.in</code>	<code>reduce.out</code>
7 15 0 0 0 10 5 10 7 9 10 6 5 -2 3 -2	2.5
7 30 0 0 0 10 5 10 7 9 10 6 5 -2 3 -2	43.0

Задача К. Ролевая игра

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

Когда у Жени бывает свободное время, он любит тратить его на свою любимую ролевую игру. Конечно же, у него есть мечта — достичь 255 уровня, то есть набрать необходимое количество очков опыта. К сожалению, у него не так много свободного времени, потому он хочет использовать его максимально эффективно и просит Вас помочь ему в этом.

Набирать очки опыта в игре можно посредством выполнения некоторых заданий. Для каждого задания известно время его выполнения, сколько очков оно приносит и список заданий, которые необходимо выполнить до того, как это задание станет доступно.

Помогите Жене выяснить, какое максимальное количество опыта, он сможет набрать за свободное время.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m ($1 \leq n \leq 16, 1 \leq m \leq 1000$) — количество заданий и количество свободного времени у Жени в минутах. В следующих n строках следуют описания заданий по одному в каждой строке в следующем формате: сначала два целых числа t_i, p_i и k_i ($1 \leq t_i, p_i \leq 1000, 0 \leq k_i < n$) — время в минутах, которое занимает выполнение i -го задания, сколько опыта получит персонаж Жени при его выполнении, и количество заданий, которые необходимо выполнить до i -го, соответственно. Далее следуют k_i различных чисел от 1 до n — номера заданий, которые необходимо выполнить до i -го.

Формат выходных данных

В первой строке выведите единственное целое число — максимальное количество опыта, которое Женья может набрать за свободное время. В следующей строке выведите номера заданий, которые необходимо выполнить для получения такого количества опыта, в том порядке, в котором их необходимо выполнять.

Примеры

<code>rpg.in</code>	<code>rpg.out</code>
2 10 5 3 1 2 5 2 0	5 2 1
3 9 4 1 0 5 2 1 1 6 3 1 1	3 1 2