

Задача А. Три единицы

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

Требуется найти количество последовательностей заданной длины из нулей и единиц, в которых не встречается трёх единиц подряд.

Формат входных данных

Входной файл содержит одно натуральное число n — длину последовательностей ($1 \leq n \leq 10^5$).

Формат выходных данных

В выходной файл выведите количество последовательностей по модулю 12345.

Примеры

<code>ones.in</code>	<code>ones.out</code>
1	2
4	13

Задача В. Плавные числа

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

Назовем натуральное число плавным, если разность любых двух его соседних цифр не превосходит по модулю единицы. Вам необходимо определить количество N -значных плавных чисел.

Формат входных данных

В единственной строке входного файла одно число N ($1 \leq N \leq 20$).

Формат выходных данных

Вывести одно число — искомое количество.

Пример

<code>smooth.in</code>	<code>smooth.out</code>
1	9

Задача С. Ядра

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

Капитан Вася всегда держит на своем корабле запас пушечных ядер для борьбы с пиратами. Так как он привык во всем поддерживать порядок, он хранит ядра в виде пирамид. Каждый из слоёв одной пирамиды является заполненным ядрами равносторонним треугольником, сторона которого содержит ровно k ядер. Сторона основания пирамиды состоит из n ядер, в следующем слое сторона состоит из $n - 1$ ядра, и так далее, пока на вершину не будет положено одно ядро (которое является равносторонним треугольником со стороной 1).

Например, пирамида размера 3 состоит из трех уровней, выглядящих так (сверху вниз):

```
  X
 X X
X X X
```

```
  X
 X X
X X X
```

Ясно, что каждый из треугольников может содержать только 1, 3, 6, 10, ... ядер. Таким образом, пирамида может содержать только 1, 4, 10, 20, ... ядер.

Вася отправляется в плавание и берет с собой ровно m ядер. Какое минимальное число пирамид требуется ему сложить из них на своем корабле?

Формат входных данных

В первой строке ввода записано количество тестов $1 \leq T \leq 20$. В последующих T строках задается количество ядер в i -м тесте $1 \leq m_i \leq 300\,000$.

Формат выходных данных

Для каждого из T тестов выведите в отдельной строке минимальное количество пирамид.

Пример

balls.in	balls.out
5	1
1	2
5	3
9	3
15	2
91	

Задача D. Кирпичи

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

Дано бесконечное количество кирпичей двух цветов — красного и синего. Красные кирпичи имеют длину 2 дециметра, а синие — 3 дециметра. Сколько различных способов выложить ряд из кирпичей длины n дециметров? Способы считаются различными, если на каком-то одинаковом расстоянии от начала ряда в них лежат кирпичи разного цвета.

Формат входных данных

В первой строке входного файла задано целое число n ($0 \leq n \leq 80$).

Формат выходных данных

В первой строке выходного файла выведите одно число — количество способов выложить ряд из кирпичей длины n дециметров.

Примеры

bricks.in	bricks.out
2	1
6	2

Задача E. Максимальная сумма

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

Дана последовательность чисел a_1, a_2, \dots, a_n . Разрешается выбрать из неё подпоследовательность $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($1 \leq i_1 \leq i_2 \leq \dots \leq i_k, 0 \leq k \leq n$) такую,

чтобы любые два индекса i_r и i_s отличались больше, чем на 1 (таким образом, в подпоследовательность не могут попасть два числа, стоящие рядом в исходной последовательности). Как осуществить этот выбор так, чтобы сумма чисел в подпоследовательности была максимально возможной?

Формат входных данных

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

Формат выходных данных

В единственной строке выведите максимально возможную сумму чисел в подпоследовательности.

Примеры

maxsum.in	maxsum.out
2 1 2	2
1 -1	0
4 -10 10 10 -10	10

Задача F. Зайчик

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

Зайчик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: если зайчик прыгнет на такую клетку, ему несдобровать. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, зайчик сможет отдохнуть и подкрепиться.

Зайчик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото и скушав как можно больше вкусной зелёной травы. Конструктивные особенности зайчика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k+1$, $k+3$ и $k+5$.

Выясните, какое максимальное количество клеток с травой сможет посетить зайчик на своём пути.

Формат входных данных

В первой строке входного файла задано число n — количество клеток ($2 \leq n \leq 1000$). Вторая строка состоит из n символов; i -ый символ соответствует i -ой клетке просеки. Символ 'w' обозначает болото, символ '.' — зелёную траву, а символ '.' соответствует клетке без каких-либо особенностей. Гарантируется, что первая и последняя клетки не содержат болот и травы.

Формат выходных данных

В первой строке выходного файла выведите одно число — максимальное количество клеток с травой, которые зайчик сможет посетить на своём пути. Если зайчику не удастся оказаться в клетке с номером n , выведите -1 .

Примеры

lepus.in	lepus.out
4 ."".	2
5 .w".."	0
9 .www.www.	-1

Задача G. Кролик

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

Как и зайчик, кролик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: прыгать в них кролик боится. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, кролик прихватит с собой пучок травы.

Кролик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото. Конструктивные особенности кролика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k + 2$, $k + 3$ и $k + 6$.

Поскольку в клетке с номером n кролика ждёт его крольчиха, главное для него — оказаться там как можно быстрее, то есть сделав как можно меньше прыжков. Если маршрутов с минимальным количеством прыжков несколько, кролик предпочтёт тот из них, на котором он посетит как можно больше клеток с травой. Если и таких маршрутов несколько, кролика устроит любой из них.

По данной карте просеки найдите оптимальный маршрут для кролика.

Формат входных данных

В первой строке входного файла задано число n — количество клеток ($2 \leq n \leq 1000$). Вторая строка состоит из n символов; i -ый символ соответствует i -ой клетке просеки. Символ 'w' обозначает болото, символ '.' — зелёную траву, а символ '.' соответствует клетке без каких-либо особенностей. Гарантируется, что первая и последняя клетки не содержат болот и травы.

Формат выходных данных

В первой строке выходного файла выведите через пробел два числа — минимальное количество прыжков k , за которое кролик может попасть из первой клетки просеки в последнюю, и максимальное количество клеток с травой t , которые он при этом сможет посетить. Во второй строке выведите через пробел $k + 1$ число — номера клеток, которые должен посетить кролик, в порядке их посещения.

Если кролику не удастся оказаться в клетке с номером n , выведите в выходной файл одно число -1 .

Примеры

rabbit.in	rabbit.out
4 ."".	1 0 1 4
5 .w".."	2 1 1 3 5
9 .www.www.	-1

Задача Н. Сколько лестниц?

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

Назовём *лестницей глины* L и *высоты* H последовательность чисел $a_0, a_1, a_2, \dots, a_L$ такую, что:

- $a_0 = 0, a_L = H$;
- для любого $0 < i \leq L$ либо $a_i = a_{i-1} + 1$, либо $a_i = a_{i-1} + 2$.

По заданным L и H найдите количество различных лестниц длины L и высоты H . Лестницы считаются различными, если соответствующие им последовательности различаются хотя бы в одном элементе.

Формат входных данных

В единственной строке ввода заданы два числа L и H через пробел ($1 \leq L, H \leq 15$).

Формат выходных данных

Выведите в единственной строке одно число — количество лестниц длины L и высоты H .

Примеры

stairs.in	stairs.out
1 1	1
2 3	2

Задача I. Ещё о лестницах

Имя входного файла: stairs2.in
Имя выходного файла: stairs2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Назовём *лестницей глины* L и *высоты* H последовательность чисел $a_0, a_1, a_2, \dots, a_L$ такую, что:

- $a_0 = 0, a_L = H$;
- для любого $0 < i \leq L$ либо $a_i = a_{i-1} + 1$, либо $a_i = a_{i-1} + 2$.

Назовём *рельефом глины* L и *высоты* H последовательность чисел $b_0, b_1, b_2, \dots, b_L$ такую, что для любого $0 \leq i \leq L$ верно $0 \leq b_i \leq H$.

Будем говорить, что *лестница* $a_0, a_1, a_2, \dots, a_L$ *ложится на рельеф* $b_0, b_1, b_2, \dots, b_L$, если для любого $0 \leq i \leq L$ верно $a_i \geq b_i$.

По заданным L, H и рельефу найдите количество различных лестниц длины L и высоты H , лежащих на этот рельеф. Лестницы считаются различными, если соответствующие им последовательности различаются хотя бы в одном элементе.

Формат входных данных

В первой строке ввода заданы два числа L и H через пробел ($1 \leq L, H \leq 20$). Во второй строке заданы $L + 1$ чисел $b_0, b_1, b_2, \dots, b_L$ через пробел — данный рельеф.

Формат выходных данных

Выведите в единственной строке одно число — количество лестниц длины L и высоты H , лежащих на данный рельеф.

Примеры

stairs2.in	stairs2.out
1 1 0 0	1
2 3 0 0 1	2
3 4 0 2 0 1	1

Задача J. Грибы

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

Маша решила навестить свою бабушку. Она взяла с собой две корзинки — одну с пирожками, а другую — пустую, для грибов, которые она хочет собрать по пути.

Для того, чтобы попасть к бабушке, Маше необходимо пройти через лес, который представляет собой прямоугольник размером $m \times n$, в некоторых клетках которого растут деревья, а в некоторых — грибы. Маша выходит из клетки $(1, 1)$ и идёт к бабушке в деревню, расположенную в клетке (m, n) .

Каждым своим ходом Маша может пойти вправо или вниз (то есть увеличить одну и только одну из своих координат на 1), если в клетке, в которой она после этого окажется, не стоит дерево. Если в обеих клетках и справа, и снизу, находятся деревья, то Маша считается заблудившейся.

Вам необходимо по данному лесу выяснить, может ли Маша дойти до бабушки, не заблудившись, и если может, то посчитать максимальное количество грибов, которое она может при этом собрать.

Формат входных данных

В первой строке входного файла находятся четыре числа m, n, g, t ($2 \leq m, n \leq 100, 0 \leq g, t \leq g + t \leq m \cdot n - 2$). В следующих g строках расположены по два числа в каждой — x и y -координаты i -го гриба. За ними следуют t строк с описаниями деревьев в аналогичном формате. Ни в какой клетке не может расти больше одного гриба, гриб и дерево одновременно, или больше одного дерева. Кроме того, в клетках $(1,1)$ и (m,n) ничего не растет.

Формат выходных данных

Если Маша может дойти до бабушки, то в первой строке выходного файла необходимо выдать максимальное количество грибов, которое она сможет при этом собрать, а в последующих $m + n - 1$ строках нужно выдать координаты клеток, последовательно посещаемых Машей, в формате $x_i y_i$, для пути, на котором достигается максимальное количество грибов. Если таких путей несколько, то разрешается выдавать любой из этих путей.

В противном случае в выходной файл должно быть выведено единственное число -1 .

Примеры

mushroom.in	mushroom.out
4 4 3 2	2
1 4	1 1
2 3	1 2
4 3	1 3
2 2	2 3
3 4	3 3
	4 3
	4 4
2 2 0 2	-1
1 2	
2 1	

Задача К. Наибольшая общая подпоследовательность

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

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычёркиванием некоторых элементов).

Формат входных данных

В первой строке входного файла записано число N — длина первой последовательности ($1 \leq N \leq 1000$).

Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 1000$).

В четвёртой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

В выходной файл требуется вывести единственное целое число: длину наибольшей общей подпоследовательности, или число 0, если такой не существует.

Пример

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

Задача L. Три последовательности

Имя входного файла: `threeseq.in`
Имя выходного файла: `threeseq.out`
Ограничение по времени: 2 секунды
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 ($1 \leq n \leq 100$), а вторая — её 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

<code>threeseq.in</code>	<code>threeseq.out</code>
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

Задача M. Поле

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

Отряду нужно пересечь прямоугольное поле размера $m \times n$ квадратов, двигаясь из левого верхнего угла в правый нижний и перемещаясь между соседними квадратами только в двух направлениях — вправо и вниз. Поле не очень ровное, но у отряда есть карта, на которой отмечена высота каждого квадрата. Опасность перехода с квадрата высоты h_1 на соседний квадрат высоты h_2 оценивается числом $|h_2 - h_1|$; опасность всех переходов в пути суммируется. Выясните, какова минимальная опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Формат входных данных

В первой строке входного файла заданы два числа m и n через пробел ($1 \leq m, n \leq 100$). В следующих n строках записано по m чисел в каждой; i -ое число j -ой из этих строк соответствует высоте квадрата (i, j) . Все высоты — целые числа в диапазоне от 1 до 100, включительно.

Формат выходных данных

Выведите в выходной файл одно число — минимальную опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Примеры

<code>field.in</code>	<code>field.out</code>
2 2 1 1 1 1	0
4 2 1 2 3 5 3 8 4 7	6
2 3 1 2 2 3 3 1	4

Задача N. Мне это больше не нужно!

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

Один отважный путешественник отправился облетать шар земной на шаре воздушном. К сожалению, его воздушный шар подхватило бурей. Шар начал падать, и путешественник приготовился было умирать, но, к счастью, на горизонте показался остров. Остров был маленький, однако стало ясно, что если выкинуть из корзины вещи, то можно до него долететь, и даже перелететь. Известно, что если путешественник выкинет вещи суммарным весом больше, чем w , то он перелетит за край острова и упадёт в бушующее море, где его ждёт неминуемая смерть. Естественно, что его не устраивает такой исход, и потому он хочет выкинуть вещей, с одной стороны, как можно большего суммарного веса (чтобы гарантированно оказаться на острове), с другой стороны, этот вес ни в коем случае не должен превосходить w . Нужно написать программу, которая бы решила бы, какие вещи нужно выкидывать. Помогите отважному путешественнику!

Формат входных данных

В первой строке входного файла заданы два целых числа: n (число вещей в корзине воздушного шара, $1 \leq n \leq 20$) и w (предельный вес, $1 \leq w \leq 10^9$). Во второй строке записаны n целых чисел, разделённые пробелами, каждое из которых описывает вес некоторой вещи, находящейся в корзине воздушного шара. Каждое из этих чисел лежит в пределах от 1 до 10^6 .

Формат выходных данных

В первой строке необходимо выдать максимально возможное значение суммарного веса w_{max} . Во второй строке необходимо вывести список номеров вещей, взяв которые, можно достигнуть такого веса. Номера должны выводиться через пробел. Если оптимальных решений несколько, разрешается выводить любое из них.

Пример

dontneed.in	dontneed.out
5 13	12
3 5 6 9 9	1 5

Задача O. Деньги

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

У вас имеется неограниченное количество монеток N заданных достоинств. Вам требуется определить, можно ли с их помощью разменять заданные K сумм денег.

Формат входных данных

В первой строке число N , далее во второй строке N чисел, задающих достоинства монеток. В третьей строке — число K , далее в четвёртой K чисел, определяющих размеры сумм. Все числа во входном файле натуральные и не превосходят 1000.

Формат выходных данных

В единственной строке K чисел — для каждой суммы 0, если её разменять нельзя, и 1, если можно.

Пример

money.in	money.out
2	1 1 0 1 1
3 5	
5	
3 6 7 11 12	