

Задача А. Плохая подстрока

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

Найдите, сколько существует строк заданной длины n , состоящих только из символов ‘a’, ‘b’ и ‘c’, и не содержащих подстроки “ab”.

Формат входных данных

Во входном файле задано n ($0 \leq n \leq 22$).

Формат выходных данных

Выведите количество таких строк.

Примеры

<code>badsubs.in</code>	<code>badsubs.out</code>
0	1
3	21
11	46368

Задача В. 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

<code>bricks.in</code>	<code>bricks.out</code>
2	1
6	2

Задача С. Один козёл

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

Лишь один козёл остался в живых после истории с кровной мстостью. Он успешно пересёк болото, но теперь он хочет вернуться домой. Для этого ему придется пересечь болото опять! Посчитайте число последовательностей ходов, которые приводят к его смерти, а также число последовательностей ходов, которые приводят к его успешному возвращению домой.

Козёл начинает свое движение из клетки $(1, 1)$ и может увеличить одну или обе своих координаты на единицу (т. е. передвинуться вниз, вправо или по диагонали вниз-вправо). Болото представляет собой прямоугольную доску $m \times n$, некоторые квадраты которой покрыты непроходимой трясиной. Если козёл попадает на квадрат с трясиной, он тут же тонет в болоте и умирает. Дом козла находится в клетке (m, n) , после того, как козел встанет на эту клетку, он считается выжившим и успешно достигшим своего дома.

Болото со всех сторон ограничено лесом, поэтому козёл не может выйти за пределы прямоугольника $m \times n$.

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных. Каждый набор начинается строкой, содержащих три целых числа m , n и k ($1 \leq m, n \leq 25$, $0 \leq k \leq mn - 2$). Следующие k строчек содержат координаты клеток, покрытых трясиной. Гарантируется, что клетки $(1, 1)$ и (m, n) трясиной не покрыты. Имейте в виду, что каждая строка во входном файле завершается переводом строки, а первый тест совпадает с тестом из примера!

Формат выходных данных

В выходном файле должно содержаться два числа для каждого из наборов входных данных. Первое число — количество корректных последовательностей ходов козла, приводящих к его смерти; второе — количество последовательностей, которые приводят к тому, что козёл успешно вернулся домой живым.

Пример

<code>goat.in</code>	<code>goat.out</code>
2 2 1	1 2
1 2	2 1
2 2 2	
1 2	
2 1	

Задача D. 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

<code>lepus.in</code>	<code>lepus.out</code>
4 ."".	2
5 .w".."	0
9 .www.www.	-1

Задача E. 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 натуральны 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

<code>money.in</code>	<code>money.out</code>
2 3 5 5 3 6 7 11 12	1 1 0 1 1

Задача F. 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 (т. е. 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 ($2 \leq m, n \leq 100$, $0 \leq g, t \leq g + t \leq mn - 2$). 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 растёт.

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 1 4 2 3 4 3 2 2 3 4	2 1 1 1 2 1 3 2 3 3 3 4 3 4 4
2 2 0 2 1 2 2 1	-1

Задача G. Гвоздики

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

На прямой дощечке вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить какие-то пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

Формат входных данных

В первой строке входного файла записано число N — количество гвоздиков ($2 \leq N \leq 100$). В следующей строке записано N чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10 000).

Формат выходных данных

В выходной файл нужно вывести единственное число — минимальную суммарную длину всех ниточек.

Пример

nails.in	nails.out
5 4 10 0 12 2	6

Задача H. 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 12 345.

Примеры

ones.in	ones.out
1	2
4	13

Задача I. 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	2 1
.w"..	1 3 5
9	-1
.www.www.	

Задача J. Спуск с горы

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

В одном из горнолыжных курортов Италии проводятся соревнования по горнолыжному спуску. Каждому спортсмену предстоит скатиться с горы на лыжах. На любом этапе спуска участник получает определённое число очков. После прохождения трассы очки суммируются. Участник, набирающий наибольшее количество очков, выигрывает. Гора представляет собой треугольник, в качестве элементов которого выступают целые числа — очки за прохождение этапа. На каждом уровне спортсмену предоставляется выбор —

двигаться вниз влево или вниз вправо. Начало спуска — в самой высокой точке горы, конец в одной из самых низких.

```
  1
 4 3
5 6 7
8 9 0 9
```

Требуется найти максимальное количество очков, которое может набрать спортсмен.

Формат входных данных

Во входном файле содержится целое число n — число этапов ($1 \leq n \leq 100$), далее n строк, каждая из которых характеризует свой уровень. В строке с номером i содержится ровно i целых чисел: a_1, a_2, \dots, a_i ($-100 \leq a_k \leq 100, 1 \leq k \leq i$) — количество очков в каждой из позиций.

Формат выходных данных

В результирующем файле должно находиться искомое целое число.

Пример

slalom.in	slalom.out
4	20
1	
4 3	
5 6 7	
8 9 0 9	

Задача K. 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

smooth.in	smooth.out
1	9