

Задача 1. Биномиальные коэффициенты

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Биномиальный коэффициент C_n^k — это количество битовых массивов длины n , в которых ровно k битов единичные.

Подробнее о биномиальных коэффициентах можно прочитать, например, в википедии: <https://ru.wikipedia.org/wiki/>

Нужно вычислить все биномиальные коэффициенты для $n \leq 1\,000$ при помощи треугольника Паскаля. Подробнее о нём можно прочитать, опять же, в википедии.

Треугольник Паскаля позволяет вычислять все биномиальные коэффициенты в порядке увеличения n , т.к.:

$$C_n^k = C_{n-1}^{k-1} + C_{n-1}^k \quad \text{при } 0 < k < n$$

После того, как все коэффициенты вычислены, нужно прочитать набор пар n и k из входного файла и выдать для каждой пары соответствующий коэффициент C_n^k .

Формат входных данных

В первой строке содержится целое число Q — количество запросов в файле ($1 \leq Q \leq 10\,000$). В каждой из следующих Q строк содержится по два целых числа n и k , для которых нужно распечатать коэффициент ($0 \leq k \leq n \leq 1\,000$).

Формат выходных данных

Нужно вывести Q вещественных чисел, по одному в строке — биномиальные коэффициенты для запросов из входном файле.

Внимание: хоть биномиальные коэффициенты и целые, они могут быть очень большими. Поэтому вычисляйте их как вещественные числа с использованием типа `double`, и распечатывайте при помощи формата `"%0.10g"` !

Пример

<code>input.txt</code>	<code>output.txt</code>
8	1
4 0	4
4 1	6
4 2	4
4 3	1
4 4	252
10 5	1.008913445e+29
100 50	2.702882409e+299
1000 500	

Пояснение к примеру

Первые пять запросов распечатывают коэффициенты для $n = 4$. Последний запрос распечатывает самый большой коэффициент, который может быть запрошен в данной задаче.

Задача 2. Японский кроссворд

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Когда Кролику нечего делать, Кролик не погружается в рефлекссию, как Пятачок, и не занимается научными исследованиями, как Винни-Пух. Он вместо этого разгадывает всякие головоломки. И сегодня на повестке дня у него японский кроссворд.

Японский кроссворд представляет собой поле из $N \times M$ клеток. При этом шифр таков: для каждой из N строк выписывается последовательность чисел, задающих размеры групп подряд идущих чёрных клеток, перечисляемых слева направо. Соседние группы чёрных клеток должны быть разделены хотя бы одной белой клеткой. Аналогично составляется шифр по вертикали для каждого из M столбцов, причём нумерация групп чёрных клеток идёт сверху вниз. Кролик любит разгадывать такие кроссворды сам, а от вас, наоборот, требуется составлять шифры по заданным полям.

Формат входных данных

В первой строке входного файла содержатся два целых числа — N и M ($1 \leq N, M \leq 10$).

В каждой из следующих N строк содержится по M символов, разделённых пробелами. Строки могут содержать только символы **0** и **1**. **0** обозначает белую клетку, а **1** — чёрную.

Формат выходных данных

Выходной файл должен содержать $N + M$ строк.

В i -й строке выходного файла должен быть шифр i -й строки кроссворда: последовательность чисел, соответствующих размерам групп чёрных клеток в этой строке, разделённых пробелами ($1 \leq i \leq N$).

Далее в j -й строке выходного файла должен быть аналогичным образом записан шифр $(j - N)$ -го столбца кроссворда ($N + 1 \leq j \leq N + M$).

Если в какой-то строке или в каком-то столбце нет черных клеток, то нужно выдавать для них соответствующую пустую строку.

Примеры

input.txt	output.txt
2 2 0 0 1 1	2 1 1
3 3 1 0 1 0 1 0 1 0 1	1 1 1 1 1 1 1 1 1 1
4 4 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1	2 2 1 2 4 3 1 1 1 2 1 2

Задача 3. Система линейных уравнений

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: разумное
Ограничение по памяти: разумное

Дана система из N линейных уравнений и N переменных:

$$a_{1,1} \cdot x_1 + a_{1,2} \cdot x_2 + \dots + a_{1,N} \cdot x_N = b_1$$

$$a_{2,1} \cdot x_1 + a_{2,2} \cdot x_2 + \dots + a_{2,N} \cdot x_N = b_2$$

\vdots

$$a_{N,1} \cdot x_1 + a_{N,2} \cdot x_2 + \dots + a_{N,N} \cdot x_N = b_N$$

Требуется решить данную систему методом Гаусса. Гарантируется, что данная система имеет единственное решение.

Формат входных данных

В первой строке задано число N — количество уравнений и переменных ($1 \leq N \leq 200$).

Следующие N строк содержат описывают систему уравнений. i -ая строка содержит по $(N+1)$ вещественному числу: коэффициенты $a_{i,1}, a_{i,2}, \dots, a_{i,N}$ и значение правой части системы b_i .

Формат выходных данных

Выведите N строк: значения переменных x_1, x_2, \dots, x_N .

Ответ будет считаться правильным, если абсолютная погрешность в каждой строке не будет превышать 10^{-3} .

Примеры

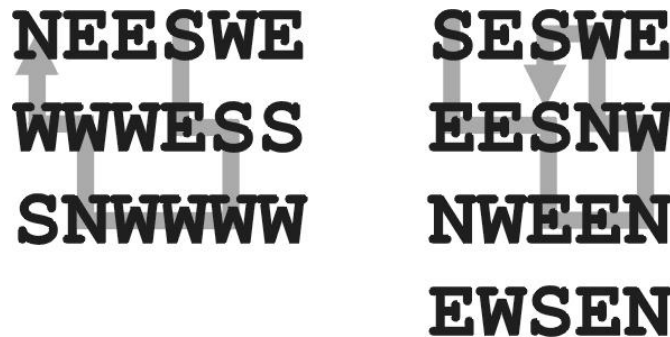
input.txt	output.txt
3	2.0000000000
2 1 -1 8	3.0000000000
-3 -1 2 -11	-1.0000000000
-2 1 2 -3	
2	-0.0000000000
0.2 0.3 0.4	1.3333333333
0.5 0.6 0.8	

Задача 4. Движение робота

Источник:	основная
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Робот был запрограммирован, чтобы следовать инструкциям в пути. Инструкции указывают направление на сетке, куда робот должен переместиться и остановиться. Возможны следующие инструкции:

- N – север (вверх по странице),
- S – юг (вниз по странице),
- E – восток (вправо по странице),
- W – запад (влево по странице).



Например, предположим, что робот начинает движение на севере (вверху) сетки (первый пример) и начинает двигаться на юг (вниз). Путь, пройденный роботом, показан на левом рисунке. Робот выполняет 10 инструкций на сетке, прежде чем покинуть ее.

Сравните с тем, что случается на другой сетке (правый рисунок): робот проходит 3 инструкции только один раз, и затем начинается цикл из 8 инструкций, из которого нет выхода.

Ваша задача состоит в том, чтобы написать программу, определяющую, сколько шагов потребуется роботу, чтобы покинуть сетку, или сколько он сделает шагов до цикла, и сколько шагов составляют цикл.

Формат входных данных

В первой строке входного файла содержатся три целых числа, разделенных пробелами: число строк в сетке, число столбцов в сетке, и номер столбца, с которого робот стартует с севера. Возможные стартовые столбцы нумеруются, начиная с единицы слева направо.

Затем идут строки направляющих инструкций. Каждая сетка имеет хотя бы 1 и не более 10 строк и столбцов инструкций. Строки инструкций состоят только из символов N, S, E и W без пробелов

Формат выходных данных

Ваша программа должна выводить одну строку, если робот, следуя некоторому числу инструкций, покидает сетку на любой из четырех сторон, или если робот следует инструкциям на некотором количестве клеток, а затем следует инструкциям на некотором количестве клеток циклически.

Примеры

input.txt
3 6 4 NEESWE WWESS SNWWW
output.txt
10 step(s) to exit

input.txt
4 5 1 SESWE EESNW NWEEN EWSEN
output.txt
3 step(s) before a loop of 8 step(s)

Пояснение к примеру

Пример входного файла отвечает двум изображенным сеткам и иллюстрирует два вида выходных файлов. За словом **step** всегда следует (**s**) даже если перед ним стоит 1.

Задача 5. Топики

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Одним из простых методов определения темы статьи в газете является выявление наиболее часто встречающихся слов в ней. Естественно, маленькие служебные слова, такие как **the**, **a** или **of** нужно игнорировать, т.к. они не влияют на содержание статьи.

Ваша задача – найти в предложенной статье наиболее часто встречающееся слово, которое не входит в список слов, которые нужно игнорировать.

Формат входных данных

В первой строке входного файла задан список слов, которые нужно игнорировать. Между словами стоит ровно один пробел. Их количество не превышает 20.

Остальные строки входного файла содержат текст, в котором нужно найти наиболее встречающееся слово. Слова не разрываются концом строки. Каждая строка начинается с нового слова. Статья заканчивается двумя звездочками (**). Регистр букв в словах нужно игнорировать. Однокоренные слова с разными окончаниями считаются разными словами. Знаки препинания тоже нужно игнорировать.

Формат выходных данных

В выходной файл необходимо вывести наиболее часто встречающееся в заданной статье слово. Это слово нужно вывести в верхнем регистре.

Пример

<code>input.txt</code>
the at in on to of is that The state finals of the Texas Computer Education Association Computer Programming Contest is today. Teams from all over the state of Texas are participating in the event. In last year's contest, each team brought their own computer.**
<code>output.txt</code>
COMPUTER

Задача 6. Шифр Шерлока Холмса

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Как известно, Шерлок Холмс использовал в своей деятельности различные способы шифрования. Опишем один из них.

Имеется $2 \cdot N$ строк, каждая из которых имеет $2 \cdot N$ символов. Также имеется квадрат, размером $2 \cdot N \times 2 \cdot N$ клеток, размер каждой клетки совпадает с размером текстового символа. В этом квадрате вырезано K клеток ($0 \leq K \leq N^2$). Квадрат является ключом к зашифрованному сообщению. Если мы наложим ключ на текст, то в вырезанных клетках получаем первую часть расшифрованного сообщения. Теперь поворачиваем ключ на 90 градусов по направлению часовой стрелки и снова накладываем на текст, получаем следующую часть сообщения, потом опять поворачиваем на 90 градусов, пока не получаем исходное положение. Таким образом, мы четыре раза накладываем квадрат на текст. В каждой позиции ключа буквы, составляющие текст, читаются по строкам сверху вниз, начиная с первой строки и заканчивая последней, в каждой строке они выбираются в порядке слева направо. Каждая расшифрованная часть текста приписывается в конец результирующей строки, пустой в начале.

Ключ будем называть **правильным**, если при повороте ключа мы открываем только новые клетки таблицы, то есть только те, которые в предыдущих положениях ключа были закрыты.

Формат входных данных

В первой строке входного файла записано целое число N ($1 \leq N \leq 125$).

Далее находится описание ключа: в $2 \cdot N$ строках записано по $2 \cdot N$ цифр (0 или 1), разделенных пробелами (если 1 – клетка вырезана).

Затем идет описание шифра: $2 \cdot N$ текстовых строк, имеющих по $2 \cdot N$ символов – букв латинского алфавита.

Формат выходных данных

В выходной файл необходимо вывести сообщение No, если во входных данных задан неправильный ключ. В противном случае необходимо вывести расшифрованный текст.

Примеры

input.txt	output.txt
1 1 0 0 0 zi am	zima
2 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 abcd abcd fbcd abcd	No

Задача 7. Поиск мест

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Группа из K друзей пошла в кино. Они пришли довольно поздно, чтобы выбирать хорошие места. Поэтому приходится искать места поближе друг к другу. Так как они все продвинутые студенты, они решили подойти к проблеме с научной точки зрения, а именно, рассмотреть задачу оптимизации.

В кинотеатре R рядов по C мест в каждом. Студентам доступна информация о занятых и свободных местах. Они хотят сесть как можно ближе друг к другу, не зависимо, в какой части зала такие места найдутся.

Чтобы установить формальный критерий, они решили минимизировать «распространение» своей группы по залу. Распространение определяется как площадь минимального прямоугольника со сторонами, параллельными местам, который содержит все купленные места. Будем считать площадью такого прямоугольника количество всех мест, которые он содержит.

Помогите им написать программу, которая решит их проблему.

Формат входных данных

Входной файл содержит несколько наборов тестов. Каждый набор состоит из нескольких строк. Первая строка набора содержит три целых положительных числа R , C и K ($1 \leq R, C \leq 300, 1 \leq K \leq R \times C$).

Каждая из следующих R строк содержит по C символов. j -й символ i -ой строки может быть либо 'X', если j -е место в i -м ряду занято, либо '.', если оно свободно.

Гарантируется, что всегда имеется не менее K свободных мест. Входной файл заканчивается строкой, содержащей три нуля, записанных через пробел.

Формат выходных данных

Для каждого набора выходной файл должен содержать строку, в которой записано целое число – минимальное распространение группы.

Пример

input.txt	output.txt
3 5 5 ...XX .X.XX XX... 5 6 6 ..X.X. .XXX.. .XX.X. .XXX.X .XX.XX 0 0 0	6 9

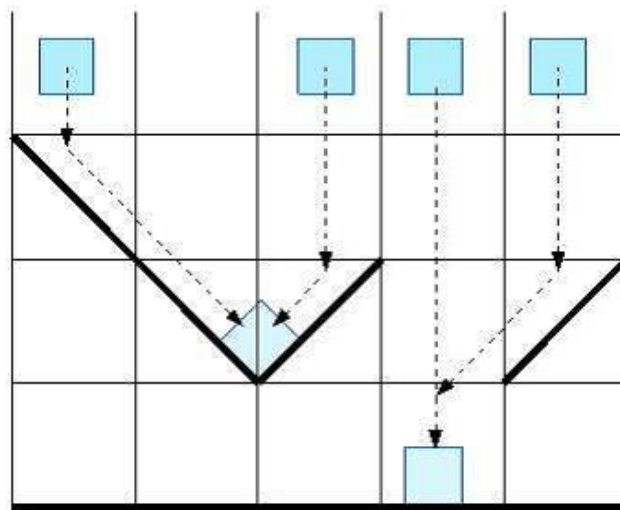
Задача 8. Водопад

Источник:	дополнительная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Ассоциация Строительных Материалов (АСМ) в очередной раз решила расширить ассортимент продаваемой продукции. На этот раз они решили заняться декорациями, а именно водопадами. Так, их новый рекламный слоган звучит: «В каждый дом по водопаду — это то, что надо!».

Бизнес, однако, этот оказался непростым, ведь надо учесть не только то, что человеку должно быть приятно любоваться своим приобретением, да ещё и сделать журчание бегущей струи бесподобным. Неудивительно, что в создании искусственного водопада есть много подводных камней, так что прежде, чем конструировать готовую версию, делается множество макетов. И лишь спустя несколько месяцев был готов первый из них: он был отображён на клетчатом листке высотой h и шириной w , в котором горки в клеточках отображались символами `'/'` и `'\'`, а символ `'_'` использовали для обозначения ровной горизонтальной поверхности.

Для начала в АСМ решили взять небольшой кубик льда, и позапускать его из разных точек. При этом кубик вёл себя следующим образом: если он находился в пустой клетке, то он падал вертикально вниз, если оказывался на горке вида `'\'`, то скатывался вниз и перемещался вправо, симметрично для горки вида `'/'`. Если он вдруг оказывался на горизонтальной поверхности, то незамедлительно останавливался. В ситуации `'\'/'` кубик также останавливается. Ниже приведены некоторые пояснительные зарисовки, которые были сделаны в ходе экспериментов:



Для того, чтобы сбалансировать звучание своего будущего творения, было принято решение посчитать количество различных точек, в которых останавливалась льдинка. Так как вас в АСМ действительно ценят, программу, выполняющую такое задание, доверили написать именно вам.

Формат входных данных

В первой строке входного файла записано два целых числа: h и w — высота и ширина клетчатого листка, соответственно ($2 \leq h, w \leq 2000$).

Далее следует h строк, в каждой из которых записано по w символов: '.' обозначает пустую клетку, 'S' обозначает пустую клетку, из которой запускали льдинку, '\' и '/' — горки, а '_' — ровные поверхности.

Гарантируется, что если льдинка, будет двигаться по описанным выше правилам, то она остановится, не выходя за пределы поля. Запуск кубика льда из одного места не оказывает влияния на запуск из другого места.

Формат выходных данных

В выходной файл требуется вывести одно целое число — количество различных точек, в которых может остановиться льдинка.

Пример

input.txt	output.txt
5 5 SSS.S S/.._ \.S.S S.../ \--/_	3
5 5 SS.SS \\./. .\./. -----	1

Задача 9. Проблемы в бухгалтерии

Источник:	дополнительная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Накануне новогодних праздников в бухгалтерии компании X скопились огромные очереди. Кто пришёл оформлять документы на отпуск, кто задержался с получением зарплаты... Работники бухгалтерии вынуждены быстро выполнять свою работу, чтобы успеть принять всех желающих. Но времени до Нового года остаётся всё меньше, а народ идёт и идёт. В связи с этим, компания обращается к вам за помощью.

Чтобы выданный бухгалтером документ имел юридическую силу, на него обязаны поставить штамп (печать). Но порой текста в документе оказывается так много, что место для печати найти достаточно проблематично. По закону, печать может лишь частично накладываться на текст документа.

В нашей задаче будем считать документ прямоугольным клетчатым листком высоты H и ширины W . Черные клетки документа обозначают занятое текстом место, белые — свободное. Штампом будем называть объект, оттиск которого нужно перенести на документ. Перед выставлением печати, разрешается передвигать штамп в любое место документа и поворачивать штамп на любой угол, кратный 90 градусов, в любую сторону. Штамп разрешается поставить в выбранном месте, только если он полностью помещается на документ (не выходит за его границы) и количество клеток документа, занятых текстом и штампом одновременно, не превосходит K .

Помогите работникам определить, можно ли выставить данный штамп на предложенный документ.

Формат входных данных

В первой строке входного файла через пробел записаны три целых числа W , H , K — размеры документа и максимальное количество накладывающихся клеток, соответственно ($3 \leq W, H, K \leq 30$).

В следующих H строках задано описание документа. Каждая строка имеет длину W . Символом '.' (точка) обозначается свободная клетка документа, символом '*' (звездочка) — клетка, занятая текстом.

Далее идет описание штампа.

В следующей строке записаны два целых числа N и M — размеры поля с рисунком штампа ($1 \leq N \leq H, 1 \leq M \leq W$).

Описание штампа состоит из N строк, каждая из которых состоит из M символов. Символом '.' обозначается клетка, не участвующая в выставлении штампа (не «окрашивающая» документ при выставлении печати), символом '*' обозначается клетка, которая должна быть «окрашена» штампом при выставлении печати. Гарантируется, что каждый из крайних столбцов и каждая из крайних строк не пусты (не состоят целиком из точек). Другими словами, на каждой грани поля, которым задаётся рисунок штампа, содержится хотя бы один символ '*'.

Формат выходных данных

В случае, если, следуя требуемым ограничениям, выставить штамп невозможно, в первую строку выходного файла выведите слово NO.

В случае, если штамп выставить возможно, в первую строку выведите слово **YES**. Во второй строке выходного файла должны быть записаны через пробел два целых числа — координаты (номер строки и номер столбца) клетки с наименьшими координатами, занятой штампом. При сравнении двух координат меньшей считается та, у которой номер строки меньше. При равенстве номеров строк меньшей считается та, у которой номер столбца меньше. Столбцы нумеруются с единицы слева направо, строки — с единицы сверху вниз.

Примеры

input.txt	output.txt
5 5 2***. .***. .***. 3 3 *** .*. ***	NO
5 5 3***. .***. .***. 3 3 *** .*. ***	YES 1 1
5 5 0 ***** *.*.* ***** ***** *.*.* 4 3 *.* *.*	YES 2 2
4 4 0 ***** *.*.* *.*.* *... 2 2 .* **	YES 3 2

Задача 10. Призрак Старого Парка

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	разумное
Ограничение по памяти:	разумное

Предлагается написать программу для решения головоломки из старой компьютерной игры «Призрак Старого Парка».

Посмотреть на головоломку в оригинале можно в этом видео:

<https://youtu.be/Ae0Nf9eXb6Q?t=565>.

Дано клеточное поле размера $N \times N$, в любой момент каждая клетка может быть белой или красной. Игрок может кликать мышкой на все клетки по собственному желанию. При нажатии на любую клетку в противоположный цвет перекрашиваются: эта клетка и все соседние с ней по стороне клетки. Обычно это пять клеток «крестом» вокруг той, на которую нажали, хотя около границы поля перекрашиваемых клеток может быть три или четыре.

Изначально все клетки поля белые. Во входном файле дана картинка, которую требуется получить. От вашей программы требуется вывести последовательность кликов, которая этого достигает.

Формат входных данных

В первой строке задано одно целое число N — размер поля ($1 \leq N \leq 20$).

В остальных N строках задано состояние поля, которое нужно получить. В каждой строке ровно N символов, каждый символ является одним из:

- точка ('.', ASCII 46) — обозначает белый цвет,
- звёздочка ('*', ASCII 42) — обозначает красный цвет.

Формат выходных данных

В первой строке нужно вывести одно целое число K — сколько кликов нужно сделать. Число K не должно превышать 10^6 . В остальных K строках должны быть заданы координаты кликов в порядке их выполнения. Для каждого клика выведите целые числа R и C — номер строки и номер столбца для той клетки в таблице, куда нужно кликнуть ($1 \leq R, C \leq N$).

Гарантируется, что заданная головоломка разрешима вообще, и разрешима при заданном ограничении на количество кликов в частности.

Примеры

input.txt	output.txt
5 ***** ***** ***** ***** *****	15 3 3 1 2 4 2 3 4 5 3 4 4 1 1 3 5 2 5 5 5 2 4 4 3 2 1 5 2 2 2
4 **.. ..*. **** .*..	4 2 2 4 4 4 3 2 1

Комментарий

Первый пример показывает, как решить оригинальную задачу из игры. Он совпадает с первым тестом. Второй пример **не** совпадает со вторым тестом.