

Задача 1. Многоэтажная дробь

Имя входного файла: input.txt или стандартный поток ввода
Имя выходного файла: output.txt или стандартный поток вывода
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Все, конечно же, знают правило сложения двух дробей. Без этого правила не получится вычислить «трёхэтажную» дробь

$$\frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + 1}}}$$

Если, например, в этой дроби $a_1 = 1$, $a_2 = 1$, $a_3 = 0$, то вся дробь будет равна $2/3$.

Чтобы построить « N -этажную» дробь, нужно задать N чисел $a_1, a_2, a_3, \dots, a_N$. Её тоже можно вычислить, записав результат в виде обыкновенной несократимой дроби p/q , где p и q – натуральные числа.

Формат входного файла

В первой строке входного файла записано натуральное число N ($1 \leq N \leq 70$). Во второй строке записаны N чисел $a_1, a_2, a_3, \dots, a_N$, разделенные пробелом, каждое из которых равно 0 или 1.

Формат выходного файла

Первая строка выходного файла должна содержать такое натуральное число p , а вторая строка – такое натуральное число q , чтобы обыкновенная дробь p/q была несократимой и равнялась заданной « N -этажной» дроби. Гарантируется, что искомые числа p и q не превосходят 10^{16} .

Примеры входных и выходных файлов

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

Пояснение к примеру

В первом примере «двухэтажная» дробь выглядит так:

$$0 + \frac{1}{1 + 1}$$

После преобразований она превратится в обыкновенную несократимую дробь $2/1$.

Во втором примере «трехэтажная» дробь выглядит так:

$$1 + \frac{1}{1 + \frac{1}{0 + 1}}$$

После преобразований она превратится в обыкновенную несократимую дробь $2/3$.

Описание подзадач и системы оценивания

Подзадача	Дополнительные ограничения	Стоимость
0	Тесты из условия	0 баллов
1	$N < 25$	50 баллов
2	$N \leq 70$	50 баллов

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 2. New Реверси

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя и Вася очень любят настольную игру «New Реверси». Правила игры:
«В игре используется квадратная доска размером $N \times N$ клеток и N^2 специальных чёрно-белых двусторонних фишек. Один из игроков играет белым цветом, другой — чёрным. Делая ход, игрок ставит фишку на клетку доски «своим» цветом вверх.

- *В начале игры доска пуста.*
- *Первый ход делают чёрные. Далее игроки ходят по очереди.*
- *Делая ход, игрок должен поставить свою фишку «своим» цветом вверх на одну из свободных клеток.*
- *Далее в каждом из 8 направлений от выбранной игроком клетки просматривается непрерывная последовательность фишек. Если эта последовательность начинается с некоторого количества идущих подряд фишек противоположного цвета, а сразу после них находится фишка такого же цвета, которым ходит игрок, либо достигается край доски, то вся эта последовательность фишек противоположного цвета переворачивается (меняет цвет)*
- *Игра прекращается, когда на доску выставлены все фишки. По окончании игры проводится подсчёт фишек каждого цвета, и игрок, чьих фишек на доске выставлено больше, объявляется победителем. В случае равенства количества фишек засчитывается ничья».*

Во время одной из партий, когда Вася ушёл за чаем, Петя чихнул, и все фишки с доски разлетелись в разные стороны. Петя не помнил ни расположение фишек, ни количество фишек каждого цвета, поэтому он просто расставил фишки на случайные позиции случайным цветом вверх.

Когда Вася вернулся, он спросил Петю, куда Петя поставил фишку в последний ход. Петя решил выбрать фишку (не важно, белую или черную), ход которой мог быть последним при данном расположении фишек, и сумма перевернутых за этот ход фишек была бы максимальна (при лучшем предыдущем состоянии игры).

Требуется определить позицию фишки, которая могла быть поставлена в последний ход, и этот ход приносил бы в лучшем случае максимальное количество перевёрнутых фишек. Если таких фишек несколько, нужно вывести позицию самой нижней из них, а при наличии нескольких "самых нижних" фишек - взять из них самую правую.

Формат входного файла

Первая строка входного файла содержит число N ($1 \leq N \leq 1000$). Следующие N строк содержат по N символов: ‘.’ (пустая клетка), ‘b’ (клетка с чёрной фишкой) или ‘w’ (клетка с белой фишкой).

Формат выходного файла

Выведите три числа, разделенные пробелом: номер столбца, номер строки искомой фишки, количество фишек, которые были перевёрнуты за ход этой фишкой. Если нет ни одной такой фишки, выведите “-1 -1 -1” (без кавычек).

Примеры входных и выходных файлов

input.txt	output.txt
4 ..b.b.. .www	2 4 2
5 ..w.b w..bb ..bb. wwwb. ww.bb	4 5 3

Пояснение к примеру 1

Рассмотрим каждую фишку:

(3, 1) – фишка на этой позиции могла быть поставлена последней, но этот ход набрал бы 0 баллов

(2, 3) – фишка на этой позиции не могла быть поставлена последней, так как, если бы её поставили последней, то белые фишки на позициях (2, 4) и (3, 4) должны были быть перевернуты

(2, 4) – фишка на этой позиции могла быть поставлена последней. Так как нам неизвестно предыдущее состояние игры, то мы можем выбрать наиболее выигрышное для нас. Таким образом, если считать, что состояние поля до хода этой фишкой имело вид

```
..b.  
....  
.b..  
..bb
```

то за этот ход было перевернуто две фишки на позициях (3, 4) и (4, 4)

(3, 4) – фишка на этой позиции могла быть поставлена в последний ход, но при этом максимум могла бы перевернуть одну фишку на позиции (4, 4)

(4, 4) - фишка на этой позиции могла быть поставлена в последний ход, и этот ход также мог принести только один балл, если предположить, что предыдущее состояние имело вид

```
..b.  
....  
.b..  
.wb.
```

Описание подзадач и системы оценивания

Подзадача	Дополнительные ограничения	Стоимость
0	Тесты из условия	0 баллов
1	$N \leq 100$	40 баллов
2	$N \leq 1000$	60 баллов

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

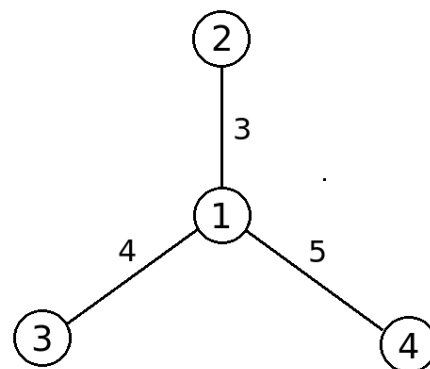
Задача 3. Дороги в стране

Имя входного файла: input.txt или стандартный поток ввода
Имя выходного файла: output.txt или стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В государстве R жители стали жаловаться на то, что их путь между городами занимает слишком много времени. Конечно же, король этого государства не может просто взять и проигнорировать эту проблему, поэтому сейчас он хочет начать разработку плана по строительству дополнительных дорог.

Король уверен, что недовольство жителей можно измерить численно и что оно равно сумме минимальных расстояний между всеми парами городов в его стране.

Например, если в стране будет 4 города и три дороги: между городами 1 и 2 длины 3, между 1 и 3 длины 4 и между 1 и 4 длины 5, как на рисунке справа, то суммарное недовольство жителей будет численно равно: $3+4+5+(3+5)+(3+4)+(4+5) = 36$.



Разработка плана по строительству дорог подразумевает два вида операций:

- 1) Построить новую дорогу между городами X и Y длиной C .
- 2) Отменить последние K совершенных операций (включая операции отмены)

До начала всех операций и после каждой король желает знать численное значение недовольства жителей. Помогите ему с этими подсчётами.

Формат входного файла

Первая строка входного файла содержит целое число N — количество городов в стране ($1 \leq N \leq 100$).

Следующие N строк содержат по N чисел, образуя матрицу $N \times N$. Число в i -й строке и j -м столбце задаёт длину дороги между городами с номерами i и j . Гарантируется, что данная матрица симметрична относительно главной диагонали и что все числа на главной диагонали — нули. Длины дорог — целые числа от 1 до 1000. «-1» означает отсутствие дороги.

Строка с номером $N+2$ содержит целое число Q — количество операций, предпринятых королём в рамках постройки плана по улучшению дорожной ситуации в своей стране.

Далее каждая из Q строк описывает очередную операцию короля:

Если это постройка дороги, то строка содержит 4 натуральных числа — $1, X, Y, C$. Где X и Y — номера городов, между которыми строится новая дорога, а C — её длина. Гарантируется, что $X \neq Y$, но при этом между двумя городами может быть построено несколько дорог. Длина новой дороги C задаётся целым числом от 1 до 1000.

Если это отмена операций, то строка содержит 2 натуральных числа — 2 и K . Где K — количество отменяемых операций. Гарантируется, что к этому моменту были совершены хотя бы K других операций.

Формат выходного файла

Выходной файл должен содержать $Q+1$ строку. Первая строка входного файла должна содержать единственное число — изначальное численное значение недовольства жителей.

Следующие Q строк должны содержать по одному целому числу — численному значению недовольства жителей после каждой операции.

Примеры входных и выходных файлов

input.txt	output.txt
4 0 3 4 5 3 0 -1 -1 4 -1 0 -1 5 -1 -1 0 5 1 2 3 5 1 2 4 3 2 1 2 2 2 2	36 34 28 34 34 28
2 0 5 5 0 5 1 1 2 6 1 1 2 3 1 1 2 2 2 2 1 1 2 4	5 5 3 2 5 4

Пояснение к примеру 1

В первом примере сначала была построена дорога между городами 2 и 3 длины 5, что уменьшило только расстояние между городами 2 и 3, которое было равно 7, а стало — 5.

Далее была построена дорога между городами 2 и 4 длины 3, что уменьшило расстояние между городами 2 и 4 (было 8, стало 3) и городами 3 и 4 (было 9, стало 8).

Далее была отменена постройка последней дороги.

Далее были отменены предыдущая отмена и постройка последней дороги.

Последним действием были отменены обе предыдущие отмены.

Описание подзадач и системы оценивания

Подзадача	Дополнительные ограничения	Стоимость
0	Тесты из условия	0 баллов
1	$N = 40, Q = 0$	15 баллов
2	$N = 40, Q = 50$	45 баллов
3	$N = 100, Q = 500$	40 баллов

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 4. Боггл

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Знаете ли Вы о такой интеллектуальной игре, как «Боггл»? «Боггл» – настольная игра, разновидность лингвистических игр (таких, как «Балда» или «Эрудит»). В «Боггл» играют обычно два человека, составляя слова из имеющихся у них букв и записывая их на бумаге.

Суть игры состоит в следующем. На стол в форме квадрата $N \times N$ выкладываются N^2 табличек, на которых написаны буквы (в классическом варианте игры $N = 4$). Засекается время, и игроки начинают составлять слова из имеющихся букв. Каждый игрок составляет свой список слов, не показывая их другим игрокам. В качестве первой буквы составляемого слова можно взять любую из имеющихся букв, а каждая следующая буква в слове должна соответствовать клетке, смежной по стороне (по горизонтали или вертикали) с предыдущей. В итоге, слову соответствует некоторая прямоугольная ломаная в исходном квадрате букв. Эта ломаная не должна иметь самопересечений и самоналожений, то есть каждую из N^2 имеющихся букв можно использовать не более одного раза. По истечении отведенного времени игроки показывают друг другу списки слов, которые они составили. За каждое правильно составленное слово игрок получает победные очки в зависимости от длины слова. У кого в сумме оказалось больше очков, тот и победил.

Начинающий программист Роман решил написать свою реализацию игры «Боггл». Так совпало, что как раз в это время Роман изучал азбуку Морзе, поэтому он решил использовать в своей реализации не русский и не английский алфавит, а алфавит Морзе (точка «.» и тире «-»). Роман уже написал часть программы, которая по входной последовательности точек и тире проверяет, существует ли осмысленное слово, кодирующееся с помощью азбуки Морзе этой последовательностью. Теперь ему нужно проверить, что написанные игроком последовательности действительно можно составить из символов исходного квадрата в соответствии с правилами игры. Помогите Роману решить эту задачу!

Формат входного файла

Первая строка входного файла содержит одно число N – размерность квадрата ($2 \leq N \leq 10$). Далее располагается исходный квадрат символов: N строк по N символов в каждой. Каждый символ – либо точка («.»), либо тире (обозначается знаком подчеркивания «_»). В следующей строке записано число K – количество последовательностей, которые написал игрок и которые нужно проверить ($1 \leq K \leq 10^5$). Далее следуют K строк, по одной последовательности в каждой строке. Каждый символ последовательности – также либо точка («.»), либо подчеркивание («_»). Длина каждой последовательности не больше 15.

Формат выходного файла

Выходной файл должен содержать K строк. Если i -ю последовательность можно составить из символов заданного квадрата в соответствии с правилами, то в i -й строке выходного файла вывести «YES», иначе в i -й строке вывести «NO».

Примеры входных и выходных файлов

<code>input.txt</code>	<code>output.txt</code>
3 ·_· -·- ... 2 .. —	YES NO

Описание подзадач и системы оценивания

Подзадача	Дополнительные ограничения	Стоимость
0	Тесты из условия	0 баллов
1	$K \leq 10$, длина каждой последовательности не больше 10	20 баллов
2	$K \leq 10$	30 баллов
3	$K \leq 10^5$	50 баллов

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

Задача 5. Накопленная сумма

Имя входного файла: `input.txt` или стандартный поток ввода
Имя выходного файла: `output.txt` или стандартный поток вывода
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Карточки пронумерованы числами $1, 2, 3, \dots, N$. На каждой из них написано какое-либо целое неотрицательное число. Миша просматривает карточки в порядке возрастания номеров. Какие-то из них он будет выбирать, а остальные – пропускать, причем после «пропущенной» карточки обязательно должна следовать «выбранная» карточка. Числа, записанные на «выбранных» карточках, Миша суммирует по следующему правилу:

- 1) число на очередной «выбранной» карточке добавляется к накопленной сумме, если предыдущая карточка тоже была «выбранной»;
- 2) число на очередной «выбранной» карточке удваивается, а затем добавляется к накопленной сумме, если предыдущая карточка была «пропущенной».

Начальная сумма равна числу, записанному на первой карточке, если она была выбрана, либо считается нулевой, если первая карточка была пропущена. Интересно, какую максимальную накопленную сумму может получить Миша, выбирая карточки по своему усмотрению и суммируя числа с «выбранных» карточек по указанному правилу?

Формат входного файла

В первой строке входного файла записано натуральное число N – количество карточек ($1 \leq N \leq 10^3$). Во второй строке содержатся N целых неотрицательных чисел, записанных на карточках с номерами $1, 2, 3, \dots, N$ соответственно. Числа разделены пробелом, каждое из них не превосходит 10^3 .

Формат выходного файла

Единственная строка выходного файла должна содержать максимальную накопленную сумму, которую может получить Миша, суммируя числа с «выбранных» им карточек по указанному выше правилу.

Примеры входных и выходных файлов

<code>input.txt</code>	<code>output.txt</code>
3 2 1 4	10
3 2 4 1	9

Пояснение к примерам

В первом примере максимальная накопленная сумма равна 10. Она получается, если выбрать первую и третью карточки.

Во втором примере максимальная накопленная сумма равна 9. Она получается, если выбрать вторую и третью карточки.

Описание подзадач и системы оценивания

Подзадача	Дополнительные ограничения	Стоимость
0	Тесты из условия	0 баллов
1	$N \leq 20$	30 баллов
2	$N \leq 1000$	70 баллов

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.