Задача 1. Просто сложение

Источник: дополнительная

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Ноль, целковый, полушка, четвертушка, осьмушка, пудовичок, медячок, серебрячок, золотничок, девятичок, десятичок.

> — Юрий Степанович Рыбников, Счёт древних русов

Степан Юрьевич, ученик выдающегося Юрия Степановича, решил на основе счета древних русов разработать систему сложения современных русов, дабы ускорить вычисления в существующих математических моделях. Сказано — сделано. И вот перед ним на столе лежит лист бумаги, на котором написана таблица суммирования новой системы. Осталось только научить компьютер складывать числа согласно этим правилам, и...

В программу закралась ошибка, и подсчет выполняется неправильно, ставя под угрозу результаты тысяч экспериментов. Нужно срочно что-то делать. Помогите Степану Юрьевичу написать новую программу, которая не будет допускать ошибок в вычислениях.

Сложение современных русов работает следующим образом. Пусть складываются два числа. Числа записаны в десятичной системе счисления. Процесс вычисления соответствует правилам вычисления чисел «в столбик».

- Числа выравниваются по младшим разрядам.
- Если числа имеют одинаковую длину, то результирующее число тоже будет иметь такую же длину, а в каждом разряде будет записана цифра из заданной таблицы, которая определяется так: если в первом числе в k-м разряде находится цифра, равная i, а во втором числе в этом же k-м разряде записана цифра, равная j, то в k-й разряд результата будет записана цифра, которая находится на пересечении i-ой строки и j-го столбца таблицы ($0 \le i, j \le 9$).
- В случае, если длины чисел различные, то в число меньшей длины слева от значащих цифр вместо ведущих нулей дописывается специальная цифра '@'. Правила сложения десятичных цифр со специальной цифрой тоже задаются табличкой.
- Правило сложения двух цифр '@' не определено

Ваша задача — вывести сумму двух заданных чисел.

Формат входных данных

В первых десяти строках входного файла записано по 10 цифр через пробел — таблица, разработанная Степаном Юрьевичем.

Сначала описываются правила сложения с 0 из первого числа, далее с 1 и так далее до 9.

В следующих двух строках задано по 10 цифр — результаты сложения цифры '@' из первого числа с десятичной цифрой из второго и, соответственно, результаты сложения десятичной цифры из первого числа с цифрой '@' из второго.

В следующей строке задано целое число Q — количество заданий ($Q \leqslant 10^3$).

Далее в Q строках заданы по два целых неотрицательных числа, которые нужно сложить. Длина каждого числа не превышает 10^3 символов.

Формат выходных данных

В выходной файл нужно вывести Q строк, содержащих по одному числу — результату сложения соответствующей пары чисел.

| input.txt | output.txt |
|---------------------|------------|
| 0 1 2 3 4 5 6 7 8 9 | 1123456780 |
| 1 2 3 4 5 6 7 8 9 0 | 2 |
| 2 3 4 5 6 7 8 9 0 1 | 6 |
| 3 4 5 6 7 8 9 0 1 2 | 07 |
| 4 5 6 7 8 9 0 1 2 3 | |
| 5 6 7 8 9 0 1 2 3 4 | |
| 6789012345 | |
| 7 8 9 0 1 2 3 4 5 6 | |
| 8 9 0 1 2 3 4 5 6 7 | |
| 9 0 1 2 3 4 5 6 7 8 | |
| 0 1 2 3 4 5 6 7 8 9 | |
| 0 1 2 3 4 5 6 7 8 9 | |
| 4 | |
| 1 1123456789 | |
| 1 1 3 3 | |
| 00 7 | |
| 8 2 4 1 2 7 2 8 0 6 | 6244 |
| 3 9 4 9 9 5 2 2 8 0 | 9673984642 |
| 8 4 5 9 8 7 8 4 6 5 | 8312297128 |
| 7 5 4 1 6 9 1 9 1 2 | 0988 |
| 8 6 6 5 5 7 2 1 5 6 | 550384 |
| 6 8 7 2 0 0 7 3 1 0 | |
| 2 7 8 3 0 7 4 6 3 8 | |
| 9 3 3 6 1 9 3 8 0 7 | |
| 6 3 8 9 0 2 1 0 9 5 | |
| 0 3 3 0 5 7 9 4 2 2 | |
| 0 5 3 9 8 0 7 4 0 6 | |
| 6 9 8 7 5 5 0 8 5 6 | |
| 5 | |
| 693 9072 | |
| 9 3962347979 | |
| 54436397 4286675697 | |
| 0 8347 | |
| 62 118221 | |

Задача 2. Поиск «суфлёра»

Источник: дополнительная

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Учитель математики Семён Владимирович в конце каждой четверти устраивает своим ученикам устный экзамен. Не секрет, что существуют неблагоразумные личности, прибегающие к различным хитростям, чтобы сдать экзамены с меньшими усилиями.

Семён Владимирович подозревает, что его ученикам помогает сдавать экзамен некий «суфлер», который пытается подорвать учебный процесс, и специально подсказывает им неправильные ответы.

Чтобы понять, существует «суфлёр» или нет, учитель вычисляет подозрительность ответа каждого ученика.

Правильный ответ и ответы всех учеников — это строки одинаковой длины. Подозрительность ответа равна длине максимальной последовательности подряд идущих символов, которые отличаются в ответе ученика от соответствующей последовательности в правильном ответе.

Формально, пусть $p_1p_2...p_N$ — правильный ответ, $s_1s_2...s_N$ — ответ ученика. Тогда подозрительность ответа равна $\max_{1\leqslant l\leqslant r\leqslant N}r-l+1$ среди таких l и r, что $p_l\neq s_l,\ p_{l+1}\neq s_{l+1},\ ...,\ p_r\neq s_r.$

Если максимальная подозрительность среди всех ответов достигает порога R, экзаменатор понимает, что «суфлёр» все-таки существует. После этого экзамен останавливается, и все ученики отправляются на пересдачу. Помогите учителю понять, существует «суфлер» или нет.

Формат входных данных

В первой строке входного файла записаны три целых числа $N,\ M,\ R$ — длина одного ответа, количество учеников, сдающих экзамен, и порог подозрительности лектора $(1\leqslant N,\ M,\ N\cdot M,\ R\leqslant 3\cdot 10^5).$

Далее следуют M+1 строк, каждая из которых состоит из N строчных английских букв. В первой строке записан правильный ответ, в следующих строках — ответы учеников.

Формат выходных данных

Для каждого ответа ученика в соответствующую строку выходного файла нужно вывести его подозрительность.

В последней строке должно быть записано слово YES, если «суфлёр» существует, и NO — иначе.

| input.txt | output.txt |
|-----------|------------|
| 4 4 3 | 1 |
| file | 3 |
| fall | 2 |
| byte | 0 |
| ball | YES |
| file | |

Задача 3. Вычисление синуса

Источник: дополнительная

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

В данной задаче нужно научиться вычислять синус. Использовать функцию **sin** из стандартной библиотеки или откуда-то ещё **запрещено**.

Подсказка: используйте ряд Тейлора.

Формат входных данных

В первой строке записано одно целое число N — количество аргументов, для которых нужно вычислить синус ($1 \le N \le 10^5$).

Далее идёт N строк, по одному вещественному числу X в каждой. Каждое число — это число, синус которого надо вычислить.

Все числа X по абсолютной величине не превышают единицу (заданы в радианах).

Формат выходных данных

Выведите N строк, в каждой строке одно вещественное число, которое равно $\sin X$ для соответствующего аргумента X из входного файла.

Рекомендуется выводить числа с помощью формата "%0.15lf", чтобы выводилось 15 знаков после десятичной точки.

Ошибка в каждом вашем ответе не должна превышать 10^{-12} .

| input.txt | output.txt |
|-------------------|-------------------|
| 5 | 0.0000000000000 |
| 0.0 | 0.5000000000000 |
| 0.523598775598298 | 0.707106781186548 |
| 0.785398163397448 | 0.866025403784439 |
| 1.047197551196597 | 1.00000000000000 |
| 1.570796326794896 | |

Задача 4. Вычисление экспоненты

Источник: дополнительная

 Имя входного файла:
 input.txt

 Имя выходного файла:
 output.txt

 Ограничение по времени:
 1 секунда

 Ограничение по памяти:
 разумное

В данной задаче нужно научиться вычислять экспоненту от заданного числа X, то есть e^X для e — натурального логарифма. Использовать функции \exp , pow и подобные из стандартной библиотеки или откуда-то ещё $\operatorname{sanpemeno}$. Лучше вообще $\operatorname{math.h}$ не подключать.

Подсказка: используйте ряд Тейлора.

Формат входных данных

В первой строке записано одно целое число N — количество аргументов, для которых нужно вычислить экспоненту $(1 \leqslant N \leqslant 10^4)$.

Далее идёт N строк, по одному вещественному числу X в каждой. Каждое число — это число, экспоненту от которого надо вычислить.

Все числа X лежат в диапазоне (-100, 100).

Формат выходных данных

Выведите N строк, в каждой строке одно вещественное число, которое равно $\exp(X) = e^X$ для соответствующего аргумента X из входного файла.

Следует выводить числа с помощью формата "%0.15g", чтобы число выводилось в экспоненциальном виде с 15 знаками после десятичной точки.

Ответ считается верным, если его относительная ошибка не превышает 10^{-12} .

| input.txt | output.txt |
|-------------------|----------------------|
| 8 | 1 |
| 0.0 | 2.71828182845905 |
| 1.0 | 0.367879441171442 |
| -1.0 | 7.38905609893065 |
| 2.0 | 0.135335283236613 |
| -2.0 | 2.68811714181614e+43 |
| 100.0 | 3.72007597602084e-44 |
| -100.0 | 1.20849583696666 |
| 0.189376476361643 | |

Задача 5. Выпуклый минимум

Источник: дополнительная

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

Массив чисел $A_0, A_1, A_2, \dots A_{n-1}$ называется выпуклым вверх, если:

$$\forall i < k < j: \qquad A_k < \frac{(j-k)A_i + (k-i)A_j}{(j-i)}$$

Дан выпуклый вверх массив A и коэффициент C. Требуется найти индекс элемента массива, на котором достигается минимум линейной функции:

$$\underset{i=0}{\operatorname{argmin}} (A_i + C \cdot i) = ?$$

Если минимальное значение достигается на нескольких элементах массива, нужно найти номер первого такого элемента.

Формат входных данных

В первой строке входного файла записано одно целое число n — размер выпуклого массива $(1 \le n \le 10^5)$.

Далее записаны элементы массива A_i (n целых чисел, $|A_i| \leq 10^{15}$).

Затем записано целое число q — количество запросов, которые нужно обработать $(1 \leqslant q \leqslant 10^5)$.

В остальных q строках записаны целые числа C_j , определяющие значения коэффициента линейной функции ($|C_i| \leq 10^9$).

Формат выходных данных

В выходной файл требуется вывести q целых чисел: для каждого коэффициента C_j , записанного во входных данных, нужно вывести номер i первого элемента A_i , на котором достигается минимум $(A_i + C \cdot i)$ при $C = C_i$.

| input.txt | output.txt |
|-------------------------|------------|
| 10 | 8 |
| 9 4 0 -2 -2 -1 1 4 8 20 | 3 |
| 8 | 5 |
| -5 | 3 |
| 1 | 0 |
| -2 | 2 |
| 0 | 2 |
| 6 | 1 |
| 3 | |
| 2 | |
| 4 | |

Пояснение к примеру

Рассмотрим коэффициент $C_2 = -2$. Выпишем значение соответствующей функции для всех элементов:

$$i=0: 9-2*0=9$$

$$i=1: 4-2*1=2$$

$$i=2: 0-2*2=-4$$

$$i=3: -2-2*3=-8$$

$$i=4: -2-2*4=-10$$

$$i=5: -1-2*5=-11$$

$$i=6: 1-2*6=-11$$

$$i=7: 4-2*7=-10$$

Минимум достигается на двух элементах i=5 и i=6, и ответом является меньший номер i=5

Комментарий

i=8: 8-2*8=-8i=9: 20-2*9=2

Представьте себе, как бы вы решали задачу, если бы вместо массива A была дана гладкая функция A(x), и нужно было бы найти минимум функции (A(x) + Cx). Задача с массивом решается точно так же, нужно лишь найти дискретный аналог для понятия производной.

Задача 6. Множество букв

Источник: дополнительная

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

Требуется прочитать множество букв и вывести это множество в виде битовой маски.

Корректно заданное множество букв начинается с открывающей круглой скобки и заканчивается закрывающей круглой скобкой. Если множество непустое, то между скобками перечислены маленькие буквы латинского алфавита, которые в него входят. Все буквы различные. Между соседними буквами всегда стоит ровно один символ запятой. Буквы могут быть перечислены в любом порядке.

В битовой маске результата должен быть установлен k-ый бит ($0 \le k < 26$) тогда и только тогда, когда k-ая буква латинского алфавита присутствует в заданном множестве.

Формат входных данных

В первой строке входного файла задано целое число Q — количество тестов ($1 \leqslant Q \leqslant 10^4$). В каждой из оставшихся Q строк записано по одному тесту.

В тесте может быть задана **любая строка**, состоящая из печатных символов ASCII (то есть символов с кодами от 32 до 126 включительно). Длина строки не превышает 100 символов, строка непустая.

Рекомендуется использовать gets или fgets для чтения строки.

Формат выходных данных

Требуется вывести ответы для всех тестов, в том же порядке, в котором эти тесты записаны во входном файле. Каждый ответ следует выводить в отдельной строке.

Если заданная в тесте строка не является корректным представлением множества букв (согласно описанию из условия), тогда нужно вывести слово Incorrect. Если же строка корректна, то нужно вывести полученную битовую маску как шестнадцатеричное число с ровно восемью цифрами. Цифры 10-15 следует выводить заглавными буквами А-F. Для вывода можно использовать формат: %08X

| input.txt | output.txt |
|---------------|------------|
| 9 | 0000000 |
| () | 00020003 |
| (a,b,r) | 00020003 |
| (r,a,b) | Incorrect |
| (a,a) | Incorrect |
| (() | Incorrect |
| (a,,b) | Incorrect |
| (A,B,R) | Incorrect |
| () a b 12345 | 03800007 |
| (z,y,x,a,b,c) | |

Задача 7. Множество чисел

Источник: дополнительная

 Имя входного файла:
 input.txt

 Имя выходного файла:
 output.txt

 Ограничение по времени:
 1 секунда

 Ограничение по памяти:
 разумное

Требуется прочитать множество чисел и вывести это множество в виде битовой маски.

Заданное множество чисел начинается с открывающей круглой скобки и заканчивается закрывающей круглой скобкой. Если множество непустое, то между скобками перечислены целые числа, которые в него входят. Все числа записаны в десятичном виде без ведущих нулей и лежат в диапазоне от 0 до 63 включительно. Все числа в множестве различные. Между соседними числами стоит ровно одна запятая. Числа могут быть перечислены в любом порядке.

В строку может быть добавлено произвольное количество пробелов в произвольных местах. При этом добавленные пробелы никогда не разрывают число на части.

В битовой маске результата должен быть установлен k-ый бит ($0 \le k < 64$) тогда и только тогда, когда число k присутствует в заданном множестве.

Формат входных данных

В первой строке входного файла задано целое число Q — количество тестов ($1 \leqslant Q \leqslant 10^4$). В каждой из оставшихся Q строк записано по одному тесту.

В каждом тесте задано множество чисел согласно описанному выше формату. Гарантируется, что множество записано корректным образом. Длина строки не превышает 200 символов.

Рекомендуется использовать gets или fgets для чтения строки.

Формат выходных данных

Требуется вывести ответы для всех тестов, в том же порядке, в котором эти тесты записаны во входном файле. Каждый ответ следует выводить в отдельной строке.

Для каждого теста нужно вывести полученную битовую маску как шестнадцатеричное число с ровно 16-ю цифрами. Цифры 10-15 следует выводить заглавными буквами А-F. Для вывода можно использовать формат: %01611X

| input.txt | output.txt |
|---------------------------------------|------------------|
| 6 | 00000000000000 |
| () | 000000000020006 |
| (1,2,17) | 000000000020006 |
| (17,1, 2) | 00000000000000 |
| () | 010000040400000E |
| (26, 56,34, 1,2, 3) (1,2,3,4,5,63) | 8000000000003E |

Задача 8. Угол в треугольнике

Источник: дополнительная

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

Требуется найти угол $\angle BAC$ в заданном треугольнике ABC.

Формат входных данных

В первой строке задано целое число Q — количество тестовых случаев $(1 \leqslant Q \leqslant 3 \cdot 10^5)$. В каждой из следующих Q строк описан один случай.

Каждый случай описывается шестью вещественными числами:

 A_x, A_y — координаты вершины угла,

 B_{x}, B_{y} — координаты другой вершины треугольника,

 C_x, C_y — координаты третьей вершины.

Все вещественные числа заданы с максимально возможной точностью и по абсолютной величине не превышают 10^3 .

Пусть M — максимум из абсолютных величин всех шести координат, заданных в тестовом случае.

Гарантируется, что $|AB|, |AC| > \frac{1}{20}M$. То есть стороны треугольника, инцидентные искомому углу, имеют довольно большую длину.

Формат выходных данных

Нужно вывести Q строк, в каждой из которых должен быть записан угол при вершине A в треугольнике ABC. Все углы нужно выводить с максимально возможной точностью, рекомендуется использовать формат "%0.20g". Углы нужно выводить в градусах, в пределах от 0 до 180 градусов включительно.

Ответ засчитывается, если он отличается от истинного менее чем на 10^{-11} .

| input.txt | output.txt |
|----------------|---------------------------|
| 8 | 36.869897645844019962 |
| 2 1 2 3 5 5 | 45 |
| 2 1 4 3 2 3 | 26.565051177077990019 |
| 3 1 3 5 2 3 | 0 |
| 0 0 1 0 10 0 | 174.28940686250035697 |
| 0 0 1 0 -10 1 | 0 |
| 7 4 3 3 3 3 | 0.00057295779511172474814 |
| 0 0 1 0 1 1e-5 | 89.999427042204885652 |
| 0 0 1 0 1e-5 1 | |

Задача 9. Длинное умножение

Источник: дополнительная

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

Во входном файле задано два целых положительных числа, одно в первой строке, второе — во второй строке. Нужно вычислить произведение этих двух чисел и вывести его в выходной файл.

Количество десятичных цифр в каждом входном числе может достигать 1000. Поэтому требуется реализовать умножение двух длинных чисел "столбиком".

| input.txt | output.txt |
|-----------|------------|
| 1991 | 60639887 |
| 30457 | |

Задача 10. Бэкап

Источник: дополнительная

Имя входного файла: input.bin Имя выходного файла: output.bin Ограничение по времени: 1 секунда Ограничение по памяти: разумное

Иммануил пользуется инновационной файловой системой собственного приготовления. Главная фишка его системы состоит в том, что для каждого файла хранится набор флагов, обозначающих тип файла. Флагов всего 30 штук, они пронумерованы по порядку до 0 до 29 включительно. Например, первые пять флагов обозначают следующее:

- флаг 0 файл является большим по размеру
- флаг 1 файл содержит текст
- флаг 2 файл связан с программированием
- флаг 3 файл содержит видео
- флаг 4 файл был создан недавно

Иммануил сделал бэкап своей файловой системы и теперь хочет проанализировать его индекс. Нужно научиться отвечать на вопрос: чему равен суммарный размер всех файлов, у которых установлен флаг k?

Формат входных данных

В первых четырёх байтах записано целое число N — количество файлов в бэкапе $(1\leqslant N\leqslant 10^3)$. Далее описываются заголовки N файлов.

Для каждого заголовка подряд записано:

- 1. Целое 32-битное число F, задающее множество установленных флагов как битовую маску $(0 \le F < 2^{30})$.
- 2. Целое 32-битное число S, обозначающее размер файла в байтах ($0 \le S < 2^{31}$).
- 3. Целое 32-битное число P, обозначающее длину имени файла ($1 \leqslant P < 100$).
- 4. Строка длиной в P байт имя файла (состоит только из букв латинского алфавита, цифр и символа точки).

Сразу после всех заголовков записано 32-битное целое Q, обозначающее количество запросов, которые нужно обработать ($1 \le Q \le 50$). После этого записаны сами запросы. Для каждого запроса указано одно 32-битное целое число X — номер флага, который сейчас интересует Иммануила ($0 \le X < 30$).

Формат выходных данных

Требуется вывести ответы на все Q запросов в порядке их перечисления во входном файле. Каждый ответ должен быть выведен как 64-битное целое число. Если в запросе указан флаг X, то ответ равен сумме размеров всех файлов бэкапа, у которых в битовой маске флагов F установлен бит номер X.

Пример

| | | | | | | | | | | j | npu | ıt.b | in | | | | | | |
|----|----|-----|----|----|----|----|----|----|----|----|-----|------------|-----|----|----|--|--|--|--|
| 06 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | D2 | 04 | 00 | 00 | 0 A | 00 | 00 | 00 | | | | |
| 72 | 65 | 61 | 64 | 6D | 65 | 2E | 74 | 78 | 74 | 19 | 00 | 00 | 00 | 8B | 78 | | | | |
| 9A | 2D | 09 | 00 | 00 | 00 | 69 | 6E | 74 | 72 | 6F | 2E | 6D | 70 | 34 | 06 | | | | |
| 00 | 00 | 00 | 01 | 09 | 00 | 00 | OA | 00 | 00 | 00 | 73 | 6F | 6C | 75 | 74 | | | | |
| 69 | 6F | 6E | 2E | 63 | 11 | 00 | 00 | 00 | 40 | 02 | 00 | 00 | 0E | 00 | 00 | | | | |
| 00 | 68 | 6F | 77 | 74 | 6F | 73 | 74 | 75 | 64 | 79 | 2E | 74 | 78 | 74 | 16 | | | | |
| 00 | 00 | 00 | F8 | 24 | 01 | 00 | 0B | 00 | 00 | 00 | 68 | 6F | 62 | 62 | 79 | | | | |
| 70 | 72 | 6F | 67 | 2E | 63 | 06 | 00 | 00 | 00 | 89 | 00 | 00 | 00 | 07 | 00 | | | | |
| 00 | 00 | 74 | 65 | 73 | 74 | 2E | 70 | 79 | 06 | 00 | 00 | 00 | 05 | 00 | 00 | | | | |
| 00 | 00 | 00 | 00 | 00 | 03 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 01 | 00 | 00 | | | | |
| 00 | 04 | 00 | 00 | 00 | | | | | | | | | | | | | | | |
| | | | | | | | | | | 0 | utp | ut. | bin | | | | | | |
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | СВ | 7A | 9A | 2D | 00 | 00 | 00 | 00 | | | | |
| 8B | 78 | 9 A | 2D | 00 | 00 | 00 | 00 | 82 | 2E | 01 | 00 | 00 | 00 | 00 | 00 | | | | |
| 54 | 33 | 01 | 00 | 00 | 00 | 00 | 00 | СЗ | 9F | 9B | 2D | 00 | 00 | 00 | 00 | | | | |

Пояснение к примеру

В файлах бэкапа используются только флаги 0, 1, 2, 3, 4, описанные в условии. Указано шесть запросов — для флагов 5, 0, 3, 2, 1 и 4 соответственно. Поскольку флаг 5 нигде не установлен, то первый ответ равен нулю.

Рассмотрим в качестве примера последний запрос для флага 4. Этот флаг установлен у видео intro.mp4 (маска флагов 19), у текстового файла howtostudy.txt (маска флагов 11), а также у файла исходного кода hobbyprog.c (маска флагов 16). Размеры этих файлов в десятичном формате равны 765098123, 576 и 75000 байтов соответственно. В сумме получается 765173699 байтов.

Задача 11. Система линейных уравнений

Источник: дополнительная

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 ($1 \le N \le 200$).

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

Формат выходных данных

Выведите N строк: значения переменных $x_1, x_2, ..., x_N$.

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

| input.txt | output.txt |
|-------------|--------------|
| 3 | 2.000000000 |
| 2 1 -1 8 | 3.000000000 |
| -3 -1 2 -11 | -1.000000000 |
| -2 1 2 -3 | |
| 2 | -0.000000000 |
| 0.2 0.3 0.4 | 1.333333333 |
| 0.5 0.6 0.8 | |

Задача 12. Призрак Старого Парка

Источник: дополнительная

Имя входного файла: input.txt
Имя выходного файла: output.txt
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 https://www.youtube.com/watch?v=Ae0Nf9eXb6Qt=565sэтом видео.

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 $(1 \le N \le 20)$.

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 $(1 \le R, C \le N)$.

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

Пример

| 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.