

Задача А. Первая задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Это будет минималистский авторский турнир. Даже в некотором смысле архаусный. Надеюсь, вам понравится, и задачи не покажутся вам слишком сложными или слишком простыми. И не пользуйтесь Интернетом!

Ваш Автор

Р. С. Это вот не слишком сложная и не слишком простая задача.

Формат входных данных

В единственной строке входного файла содержатся два целых числа через пробел: n и m ($0 \leq n \leq 10^9$, m может принимать одно из десяти значений: 1, 2, 3, 5, 7, 11, 13, 17 или 1000000007).

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное натуральное число — минимальное из натуральных чисел, которые при возведении в степень n делятся на m , или слово «ABSENT» без кавычек, если такого числа не существует.

Примеры

input.txt	output.txt
2 2	2
3 3	3

Замечание

Люблю абсент.

Задача В. Ещё одна

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Это первая задача, которая мне пришла в голову. Но в этом наборе она будет второй. Я принял такое решение. Иногда принять решение довольно сложно, но иногда сложнее бывает его найти. Например, не так легко найти количество простых чисел, не превышающих числа r . Особенно если под рукой нет компьютера, а $r = 10^9$. Если вы уже бросились искать, то хочу вас огорчить: его уже давно нашли. Неинтересно искать решение уже решённой задачи. Лучше найдите не количество простых чисел, а количество чисел, взаимно простых с n .

Решительный Автор

P. S. Два числа взаимно просты, если у них нет общих делителей, отличных от единицы. На всякий случай напишу об этом ещё раз в выходных данных. Кто читает условие, в конце-то концов?

Формат входных данных

В единственной строке входного файла содержатся два целых числа через пробел: n и r ($1 \leq n, r \leq 10^9$) — величины, описанные в условии задачи и в выходных данных. Думаю, не стоит их тут описывать ещё раз.

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное целое число — количество натуральных чисел, не превышающих r и не имеющих с n общих делителей, отличных от единицы.

Примеры

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

Замечание

Во втором примере число 9 не является простым, но является взаимно простым с числом 10, а вот число 2 является простым, но не является взаимно простым с числом 10. Забавно, не правда ли? Как-нибудь сделаю такую задачу.

Задача С. Неутешительная задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Говорят, на каждом соревновании по программированию должна быть задача, которую решают все участники этого соревнования. На моей памяти ни разу такого не было. Надеюсь, на сей раз всё исправится. Если вам нравится слово «Accepted» без кавычек, то эта задача для вас. И даже если с кавычками.

Итак, одна улитка ползла к вершине горы Фудзи. Днём она поднималась на n метров вверх, а ночью засыпала и во время сна сползала на m метров вниз. Спрашивается в задаче: на какой день она доползёт до вершины, если высота Фудзи 3776 метров?

Добрый Автор

P. S. Склон, конечно, нужно считать вертикальным. В первый день улитка начинает ползти с подножья горы.

Формат входных данных

В единственной строке входного файла содержатся два целых числа через пробел: n и m ($0 \leq n, m \leq 10^9$) — скорости подъёма и спуска улитки соответственно.

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное целое число — количество дней, которое улитка будет ползти до вершины, или слово «NEVER» без кавычек, если она никогда не доберётся до своей цели.

Примеры

<code>input.txt</code>	<code>output.txt</code>
5 4	3772
100 200	NEVER

Замечание

Знаете ли вы, что Фудзи — самая высокая гора в Японии. На её вершине соседствуют действующий синтоистский храм и метеорологическая станция, а у подножья находится лес Аокигахара, печально известный тем, что в XIX веке бедные японские семьи привозили и оставляли в этом лесу на верную гибель своих стариков и детей, которых не могли прокормить. Фудзи является действующим вулканом, но не существует ни одного изображения его извержения. Вообще-то я не знаток японской культуры, а просто только что вычитал всё это в Википедии. Никто же не читает примечания, правда?

Задача D. Робот в лабиринте

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

В детстве я любил роботов и лабиринты. Я строил лабиринты из кубиков и запускал в них роботов. Для каждого робота я составлял программу путешествия: это была строчка букв «Л», «П», «В» и «Н», которые обозначали соответственно шаги влево, вправо, вверх и вниз в некоторой дискретной системе координат с квадратами размером в один кубик. Жаль, мои роботы были игрушечные и мне приходилось исполнять эти программы за них. Я называл программу корректной, если она не приводила к врезанию в стену или выходу за пределы лабиринта. Как-то раз я построил очередной лабиринт, поместил в него робота и задумался, сколько всего существует корректных программ, которые приведут этого робота на его изначальное место. Я быстро догадался, что их бесконечно много. Стоило бы на этом остановиться, но я задумался, сколько корректных программ, не превышающих длины l , оставят робота на месте. Именно такую задачу вам и придётся сейчас решить. Лабиринт представляет собой прямоугольное поле размером n на m квадратов. Каждый квадрат может быть либо стеной, либо проходом.

Повзрослевший Автор

P. S. Интересно, в какие игры вы играли в детстве.

Формат входных данных

В первой строке входного файла содержатся три целых числа: n , m и l ($1 \leq n, m \leq 100, 1 \leq l \leq 30$) — размеры лабиринта и максимальное количество команд в программе. Далее в n строках содержатся по m символов — карта лабиринта. Символы имеют следующее значение:

- «.» — проход;
- «=» — стена;
- «*» — начальное положение робота.

Гарантируется, что на карте имеется единственное начальное положение робота.

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное целое число — количество программ из не более l команд, которые возвращают робота на исходную позицию. Пустая программа программой не считается.

Примеры

input.txt	output.txt
3 5 3 ===== =.*.= =.*==	2
4 5 4 ..=.. ..=.. =.*.= =...=	22

Замечание

Во втором примере искомые программы: ЛП, ПЛ, НВ, ЛВНП, ЛНПВ, ЛНВП, ЛПЛП, ЛППЛ, ЛПНВ, ПВНЛ, ПНЛВ, ПНВЛ, ПЛПЛ, ПЛЛП, ПЛНВ, НЛВП, НЛПВ, НПВЛ, НПЛВ, НВЛП, НВПЛ, НВНВ.

Задача Е. О дружбе

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	128 мегабайт

Я заметил, что довольно часто большие дружные компании держатся на самом деле всего на нескольких людях. Если убрать такого человека (например, если он уезжает куда-то далеко или умирает), то большая компания сразу же распадается на несколько дружных групп поменьше. Я не слишком много понимаю в человеческих отношениях, но думаю, что понимаю, почему так происходит.

На самом деле, два человека общаются между собой, если у них в достаточной мере совпадают интересы. Будем называть таких двух людей настоящими друзьями. Если же этого не происходит, то они могут оставаться в дружеских отношениях просто потому, что у них имеется общий настоящий друг или у одного из них есть настоящий друг, у которого есть настоящий друг и тот, в свою очередь, настоящий друг другого. Тогда если между такими друзьями больше не останется настоящих друзей, то они более не станут дружить друг с другом. Так и распадаются большие дружные группы.

Будем называть человека душой компании, если его удаление из компании ведёт к её распаду на более мелкие группы. Напомню, что компания остаётся единой, если любые два человека в ней являются друзьями, хотя бы даже и не настоящими. В этой задаче вам нужно найти в компании из n человек с известными парами настоящих друзей все души этой компании. Чтобы было проще, вместо имён у людей будут номера.

Одинокий Автор

P. S. Надеюсь, не слишком сумбурно.

Формат входных данных

В первой строке входного файла содержатся два целых числа: n и m ($2 \leq n \leq 10^4$, $1 \leq m \leq 10^5$) — количество друзей в компании и количество пар настоящих друзей. Далее в m строках содержатся пары целых чисел через пробел: x_i и y_i ($1 \leq x_i, y_i \leq n$) — пары настоящих друзей. Гарантируется, что компания действительно дружная и что никакая пара настоящих друзей не повторяется дважды.

Формат выходных данных

В первой строке входного файла должно быть записано единственное целое число — количество душ компании. Далее во второй строке через пробел должны быть перечислены номера людей, являющихся душами компании, в возрастающем порядке.

Примеры

input.txt	output.txt
5 6 1 3 1 5 2 4 2 5 3 5 4 5	1 5
10 11 1 2 1 3 2 4 3 4 4 5 4 6 5 7 6 7 4 9 8 9 9 10	2 4 9

Замечание

Во втором примере, удаление как четвёртого, так и девятого человека приводит к распаду компании на три группы.

Задача F. Мастер угадывания цифр

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Эту задачу может решить только один человек. Я сам. Потому что в ней нужно угадывать цифры, которые я загадал на каждый тест. Лучше даже не беритесь её решать. Все ваши усилия пропадут даром. Должна же быть хоть одна нерешаемая задача.

Поскольку сам сочинить достаточно случайную последовательность цифр я не могу, я использовал автоматический нелинейный генератор. Я даже расскажу вам, как он работает. Он просто сопоставляет номеру теста k число $g(k) = n^k + c$ и берёт последнюю цифру этого числа. Но вам никогда не удастся реализовать его, потому что я ни за что в жизни не скажу вам числа n и c , которые я использовал.

Хитрый Автор

P. S. Мва-ха-ха-ха-ха.

Формат входных данных

В единственной строке входного файла содержится единственное целое число k ($1 \leq k \leq t$) — номер теста. t — это количество тестов. Так я вам его и сказал.

Формат выходных данных

В единственной строке выходного файла должна быть записана единственная цифра, которую я загадал.

Примеры

<code>input.txt</code>	<code>output.txt</code>
1	7
2	3

Замечание

Даже не пытайтесь использовать рандом.

Задача G. Баян

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Один мой бородатый товарищ очень хорошо играет на баяне. Особенно классическую музыку. Как-то я пошёл к нему в гости послушать его замечательную игру. Легко нашёл нужный дом, но он оказался очень высоким и длинным, а табличек с номерами квартир на подъездах не было, и лифты не работали, так что я не знал, в какой подъезд мне надо войти и на какой этаж подняться. Я знал, что мой друг живёт в квартире k . Я насчитал в доме n этажей и, зайдя в один из подъездов, определил, что на каждом этаже имеется по m квартир. Оставалось только определить подъезд и этаж, потому что бегать по этажам и подъездам очень не хотелось. Утомительное это занятие.

Неспортивный Автор

P. S. В конце концов всё же оказалось, что это был не тот дом.

Формат входных данных

В единственной строке входного файла содержатся три целых числа через пробел: n , m и k ($1 \leq n, m, k \leq 10^9$) — количество этажей, количество квартир на этаже и номер нужной квартиры соответственно.

Формат выходных данных

В единственной строке выходного файла должны быть записаны два целых числа через пробел — номер подъезда, в который нужно войти, и номер этажа, на который нужно подняться, чтобы найти квартиру k .

Примеры

<code>input.txt</code>	<code>output.txt</code>
3 4 10	1 3
5 2 20	2 5

Замечание

Вы же никогда не решали ничего похожего, правда?

Задача Н. Без имени

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Мне было лень придумывать легенду к этой задаче. Честное слово, лень.

Ленивый Автор

Р. С. Это же просто отбор в Саратов, правда? Кого волнует, что у задачи нет легенды?

Формат входных данных

В первой строке входного файла содержатся три целых числа: x_1 , y_1 и r_1 ($-10^3 \leq x_1, y_1 \leq 10^3, 1 \leq r_1 \leq 10^3$) — координаты центра первой окружности и её радиус. Далее во второй строке аналогичным образом содержатся три целых числа: x_2 , y_2 и r_2 ($-10^3 \leq x_2, y_2 \leq 10^3, 1 \leq r_2 \leq 10^3$) — координаты центра и радиус второй окружности.

Формат выходных данных

В первой строке выходного файла должно быть записано единственное целое число n — количество точек пересечения двух данных окружностей. Далее в n строках через пробел должны быть записаны пары вещественных чисел — координаты очередной точки пересечения, с абсолютной или относительной погрешностью не более 10^{-6} . Точки можно выводить в любом порядке. Если их бесконечно много, то в единственной строке выходного файла должно быть записано единственное слово «MANY» без кавычек.

Примеры

input.txt	output.txt
0 0 200 300 0 200	2 150.0000000 132.2875656 150.0000000 -132.2875656
0 0 1 0 2 1	1 0.0000000 1.0000000

Замечание

Я всегда буду делать задачи с легендами. Я всегда буду делать задачи с легендами. Я всегда буду делать задачи с легендами.

Задача I. Произведения

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Поймите меня правильно, мне действительно нужна была ещё одна задача. Я просто вынужден был это сделать. Я украл эту задачу с одной школьной олимпиады. Не скажу, с какой именно, а то вы будете сдавать мне решения жюри. К тому же это будет несправедливо к другим олимпиадам, на которых могла быть точно такая же задача. В общем, ещё раз простите. Если вы уже много раз решали эту задачу, то вы наверняка признаетесь в этом членам жюри.

Наивный Автор

Р. С. Чуть не забыл условие. Если что, условия нормальных задач выглядят именно так.

Петя участвует в конкурсе юных любителей головоломок. В последнем туре участникам предложена следующая задача:

Дана квадратная таблица размером $n \times n$. Рядом с каждым столбцом и каждой строкой написано целое положительное число. Пусть около j -го столбца написано число x_j , а около i -й строки — число y_i . Для победы в конкурсе необходимо в некоторых клетках таблицы расположить натуральные числа так, чтобы выполнялись следующие условия:

1. Все эти числа должны быть различными.
2. В каждом столбце и каждой строке должно быть ровно 2 числа.
3. Произведение чисел j -го столбца должно быть равно x_j .
4. Произведение чисел i -й строки должно быть равно y_i .

Рисунок показывает пример решения такой задачи.

6	3				18
9		1			9
		12		11	132
			4	8	32
	2		5		10
54	6	12	20	88	

Помогите Пете!

Формат входных данных

В первой строке входного файла содержится единственное целое число n ($2 \leq n \leq 10$) — размер таблицы. Далее во второй строке содержатся n целых чисел через пробел: x_j ($1 \leq x_j \leq 1000$) — произведения в столбцах. Далее в третьей строке аналогичным образом содержатся n целых чисел через пробел: y_i ($1 \leq y_i \leq 1000$) — произведения в строках. Гарантируется, что существует таблица с такими значениями произведений, удовлетворяющая прочим условиям задачи.

Формат выходных данных

В выходном файле должны быть записаны n строк по n целых чисел через пробел в каждой — результирующая таблица. В i -й строке на j -й позиции должно быть записано поставленное в соответствующую ячейку таблицы натуральное число, либо число 0, если эта ячейка осталась пустой. Если решений несколько, можно вывести любое из них.

Примеры

input.txt	output.txt
2	1 3
2 12	2 4
3 8	
3	1 2 0
5 8 18	5 0 6
2 30 12	0 4 3

Замечание

Мучает совесть.

Задача J. Последняя задача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Ну вот и всё. Это последняя задача, которую я смог придумать. Это задача об ошибках. Надеюсь, я не наделал слишком много ошибок в текстах задач, в тестах и в решениях жюри, а то ведь я делаю очень много ошибок ещё со школы. Вот и во входной файл к этой задаче я хотел вписать две одинаковые строки s_1 . Я набрал строку s_1 в первый раз, а потом хотел написать заново, чтобы избежать страшных ошибок копирования. Но при списывании я всё равно ошибался, и у меня получилась другая строка s_2 . Ваша задача — определить количество ошибок, которые я сделал.

Вот и всё. Удачного вам решения задач. Тренируйтесь больше, верьте в себя и вы обязательно дойдёте до финала чемпионата мира. Играйте и выигрывайте.

Пафосный Автор

P. S. При списывании я обычно делаю два вида ошибок:

- пропуск символа;
- замена одного символа другим.

Вам следует отыскать минимально возможное количество ошибок. В конце концов, не так уж и много я их делаю.

Формат входных данных

В первой строке входного файла содержится непустая строка s_1 , состоящая из маленьких букв латинского алфавита и не превышающая по длине 1024 символов — строка, которую я должен был написать дважды. Далее во второй строке содержится непустая строка s_2 — строка s_1 , записанная с ошибками. Гарантируется, что s_2 действительно получается из s_1 описанным способом.

Формат выходных данных

В единственной строке выходного файла должно быть записано единственное целое число — минимальное количество ошибок, которое я сделал при списывании.

Примеры

<code>input.txt</code>	<code>output.txt</code>
<code>flash</code> <code>flesh</code>	1
<code>bread</code> <code>beer</code>	3

Замечание

Хорошо, когда ошибки можно исправить.