

Содержание

| | |
|---|----|
| Задача А. Два числа [0.5 sec, 256 mb] | 2 |
| Задача В. Перестановки [0.5 sec, 256 mb] | 3 |
| Задача С. Покрытие доминошками [0.5 sec, 256 mb] | 4 |
| Задача D. Лишние пробелы [0.5 sec, 256 mb] | 5 |
| Задача Е. Самая большая буква [0.5 sec, 256 mb] | 6 |
| Задача F. Ненокку (простая) [0.5 sec, 256 mb] | 7 |
| Задача G. Любители Кошек [0.5 sec, 256 mb] | 8 |
| Задача H. Дерево [0.5 sec, 256 mb] | 9 |
| Задача I. Расстояние между вершинами [0.5 sec, 256 mb] | 10 |
| Задача J. Минимальное покрытие [0.5 sec, 256 mb] | 11 |
| Задача K. Тестирующая система [0.5 sec, 256 mb] | 12 |
| Задача L. Расстояние от точки до прямой [0.5 sec, 256 mb] | 13 |
| Задача M. Произведение двух [0.5 sec, 256 mb] | 14 |
| Задача N. Восстановление [0.5 sec, 256 mb] | 15 |
| Задача O. Миллиардеры [0.5 sec, 256 mb] | 16 |
| Задача P. К-я строка [0.5 sec, 256 mb] | 17 |

Задача А. Два числа [0.5 sec, 256 mb]

Даны два целых числа A и B ($1 \leq A, B \leq 100$). Найдите два таких целых числа X и Y , что выполнено равенство $AX + BY = 1$.

Формат входных данных

Во входном файле записаны два числа A и B , разделённые пробелом.

Формат выходных данных

В выходной файл выведите два числа X и Y , разделённые пробелом. Требуется, чтобы выполнялись неравенства $|X| \leq 10\,000$, $|Y| \leq 10\,000$. Если правильных ответов несколько, разрешается вывести любой из них. Если же таких чисел не существует, выведите вместо них два нуля.

Примеры

| stdin | stdout |
|--------|------------|
| 2 3 | 2 -1 |
| 4 6 | 0 0 |
| 100 51 | -5075 9951 |

Задача В. Перестановки [0.5 sec, 256 mb]

Во входном файле задано число n ($1 \leq n \leq 8$). Выведите в выходной файл в лексикографическом порядке все перестановки чисел от 1 до n .

Пример

| stdin | stdout |
|-------|--|
| 3 | 1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1 |

Задача С. Покрытие доминошками [0.5 sec, 256 mb]

Это простая задача на перебор!

Сколько способов покрыть прямоугольник $W \times H$ доминошками?

Каждая клетка должна быть покрыта ровно одной доминошкой.

Формат входных данных

Числа W и H . Ограничения: $W \cdot H \leq 50$, кроме этого $1 \leq W, H$.

Формат выходных данных

Число способов покрыть прямоугольник доминошками.

Пример

| stdin | stdout |
|-------|--------|
| 3 2 | 3 |

Задача D. Лишние пробелы [0.5 сек, 256 mb]

Дана строка. Напишите программу, которая удалит из этой строки все лишние пробелы. Пробел будем считать лишним, если:

1. он находится в самом начале строки, до самого первого слова;
2. он находится в конце строки, после самого последнего слова;
3. несколько пробелов расположены между двумя словами (проще говоря, если слова разделены более чем одним пробелом, тогда все пробелы кроме одного — лишние).

Формат входных данных

Во входном файле записана строка, длина которой не превышает 200 символов. Строка содержит только маленькие латинские буквы и пробелы.

Формат выходных данных

Выведите в выходной файл эту строку без лишних пробелов.

Примеры

| stdin | stdout |
|------------|------------|
| first test | first test |

Задача Е. Самая большая буква [0.5 sec, 256 mb]

Дана строка. Найдите в ней букву, алфавитный номер которой — самый большой среди букв этой строки.

Формат входных данных

В первой строке входного файла задана строка длины от 1 до 100 символов, включительно. Строка состоит из строчных букв латинского алфавита.

Формат выходных данных

В первой строке выходного файла выведите одну букву — ту букву строки, алфавитный номер которой максимален.

Примеры

| stdin | stdout |
|-------|--------|
| aab | b |
| dabux | y |

Задача F. Ненокку (простая) [0.5 сек, 256 mb]

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово “книга”. Но он не любит читать книги (он лучше ползает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

Формат входных данных

В каждой строчке входного файла записано одна из двух записей.

1. ? <слово> (<слово> - это набор не более 50 латинских символов);
2. A <текст> (<текст> - это набор не более 1024 латинских символов).

1 означает просьбу проверить существование подстроки <слово> в произведение.

2 означает добавление в произведение <текст>.

Число запросов — не более 30. Входной файл содержит не более 1 килобайта.

Формат выходных данных

Выведите на каждую строчку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

Пример

| stdin | stdout |
|-------------|--------|
| ? love | NO |
| ? is | NO |
| A Loveis | YES |
| ? love | NO |
| ? WHO | YES |
| A Whoareyou | |
| ? is | |

Задача G. Любители Кошек [0.5 sec, 256 mb]

В университетском клубе любителей кошек зарегистрировано n членов. Естественно, что некоторые из членов клуба знакомы друг с другом. Нужно сосчитать, сколькими способами можно выбрать из них троих, которые могли бы свободно общаться (то есть, любые два из которых знакомы между собой).

Формат входных данных

В первой строке входного файла заданы числа n и m ($1 \leq n \leq 1000$, $1 \leq m \leq 30\,000$), где m обозначает общее число знакомств. В последующих m строках идут пары чисел a_i b_i , обозначающие, что a_i знаком с b_i . Информация об одном знакомстве может быть записана несколько раз, причем даже в разном порядке (как (x, y) , так и (y, x)).

Формат выходных данных

В выходной файл необходимо вывести количество способов выбрать троих попарно знакомых друг с другом людей из клуба.

Пример

| stdin | stdout |
|--------------------------|--------|
| 3 3 1 2 2 3 3 1 | 1 |

Задача Н. Дерево [0.5 сек, 256 mb]

Дан неориентированный граф. Проверьте, является ли он деревом.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и m — количество вершин и рёбер в графе, соответственно ($1 \leq n \leq 100$). В следующих m строках заданы рёбра; i -я из этих строк содержит два целых числа u_i и v_i через пробел — номера концов i -го ребра ($1 \leq u_i, v_i \leq n$). Граф не содержит петель и кратных рёбер.

Формат выходных данных

В первой строке выходного файла выведите “YES”, если граф является деревом, и “NO” в противном случае.

Примеры

| stdin | stdout |
|--------------------------|--------|
| 3 2 1 2 1 3 | YES |
| 3 3 1 2 2 3 3 1 | NO |

Задача I. Расстояние между вершинами [0.5 sec, 256 mb]

Коль Дейкстру́ писать без кучи,
То тайм-лимит ты получишь...
А в совсем крутой задаче
Юзай кучу Фибоначчи!

Спектакль преподавателей
ЛКШ.июль-2007

Дан взвешенный неориентированный граф. Требуется найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. Вторая строка входного файла содержит натуральные числа s и t — номера вершин, длину пути между которыми требуется найти ($1 \leq s, t \leq n$, $s \neq t$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i — номера концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100$).

$n \leq 100\,000$, $m \leq 200\,000$.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами s и t .

Если путь из s в t не существует, выведите -1.

Пример

| stdin | stdout |
|--|--------|
| 4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4 | 3 |

Задача J. Минимальное покрытие [0.5 sec, 256 mb]

На прямой задано некоторое множество отрезков с целочисленными координатами концов $[L_i, R_i]$. Выберите среди данного множества подмножество отрезков, целиком покрывающее отрезок $[0, M]$, (M — натуральное число), содержащее наименьшее число отрезков.

Формат входных данных

В первой строке указана константа M ($1 \leq M \leq 5\,000$). В каждой последующей строке записана пара чисел L_i и R_i ($|L_i|, |R_i| \leq 50\,000$), задающая координаты левого и правого концов отрезков. Список завершается парой нулей. Общее число отрезков не превышает 100 000.

Формат выходных данных

В первой строке выходного файла выведите минимальное число отрезков, необходимое для покрытия отрезка $[0, M]$. Далее выведите список покрывающего подмножества, упорядоченный по возрастанию координат левых концов отрезков. Список отрезков выводится в том же формате, что и во входе. Завершающие два нуля выводить не нужно.

Если покрытие отрезка $[0, M]$ исходным множеством отрезков $[L_i, R_i]$ невозможно, то следует вывести единственную фразу “No solution”.

Примеры

| stdin | stdout |
|----------------------------------|-------------|
| 1 -1 0 -5 -3 2 5 0 0 | No solution |
| 1 -1 0 0 1 0 0 | 1 0 1 |

Задача К. Тестирующая система [0.5 сек, 256 mb]

Юный программист Саша написал свою первую тестирующую систему. Он так обрадовался тому, что она скомпилировалась, что решил пригласить школьных друзей на свой собственный контест.

Но в конце тура выяснилось, что система не умеет сортировать команды в таблице результатов. Помогите Саше реализовать эту сортировку.

Команды упорядочиваются по правилам ACM:

- по количеству решённых задач в порядке убывания;
- при равенстве количества решённых задач — по штрафному времени в порядке возрастания;
- при прочих равных — по номеру команды в порядке возрастания.

Формат входных данных

Первая строка содержит натуральное число n ($1 \leq n \leq 100\,000$) — количество команд, участвующих в контесте. В i -й из следующих n строк записано количество решенных задач S ($0 \leq S \leq 100$) и штрафное время T ($0 \leq T \leq 100\,000$) команды с номером i .

Формат выходных данных

В выходной файл выведите n чисел — номера команд в отсортированном порядке.

Пример

| stdin | stdout |
|-------|-----------|
| 5 | 5 2 1 3 4 |
| 3 50 | |
| 5 720 | |
| 1 7 | |
| 0 0 | |
| 8 500 | |

Задача L. Расстояние от точки до прямой [0.5 sec, 256 mb]

Найдите расстояние от заданной точки до заданной прямой.

Формат входных данных

Пять целых чисел — координаты точки и коэффициенты A , B и C нормального уравнения прямой.

Формат выходных данных

Одно число — расстояние от точки до прямой с точностью не менее 10^{-6} .

Пример

| stdin | stdout |
|------------|---------|
| 1 1 1 1 -1 | 0.70711 |

Задача М. Произведение двух [0.5 sec, 256 mb]

Формат входных данных

Входной файл состоит из двух целых чисел A и B , не превосходящих по модулю 10^9 .

Формат выходных данных

Программа должна выдавать в выходной файл единственное число — произведение чисел A и B .

Пример

| stdin | stdout |
|-------|--------|
| 2 2 | 4 |

Задача N. Восстановление [0.5 sec, 256 mb]

Денис обнаружил ошибку в своей программе, которая удаляет все символы из строки кроме "(" и ")". Оказывается, некоторые символы заменяются на что-то нечитаемое.

Теперь его заинтересовал вопрос, сколько различных правильных скобочных последовательностей длины $2n$ могут являться результатом исправленного алгоритма, то есть не будут противоречить данным, которые он таки не потерял.

Формат входных данных

Единственная строка входного файла содержит строку из круглых скобок и знаков вопроса, где вопросами обозначены утраченные символы. Длина строки не превосходит 10 000, но может быть нечетной.

Формат выходных данных

Выведите одно число — количество различных скобочных последовательностей, удовлетворяющих шаблону Дениса, по модулю $10^9 + 7$.

Пример

| stdin | stdout |
|---------|--------|
| (??()?) | 2 |

Задача О. Миллиардеры [0.5 сек, 256 mb]

Возможно, вы знаете, что из всех городов мира больше всего миллиардеров живёт в Москве. Но, поскольку работа миллиардера подразумевает частые перемещения по всему свету, в определённые дни какой-то другой город может занимать первую строчку в таком рейтинге. Ваши приятели из ФСБ, ФБР, MI5 и Шин Бет скинули вам списки перемещений всех миллиардеров за последнее время. Ваш работодатель просит посчитать, сколько дней в течение этого периода каждый из городов мира был первым по общей сумме денег миллиардеров, находящихся в нём.

Формат входных данных

В первой строке записано число n — количество миллиардеров ($1 \leq n \leq 10\,000$). Каждая из следующих n строк содержит данные на определённого человека: его имя, название города, где он находился в первый день данного периода, и размер состояния. В следующей строке записаны два числа: m — количество дней, о которых есть данные ($1 \leq m \leq 50\,000$), k — количество зарегистрированных перемещений миллиардеров ($0 \leq k \leq 50\,000$). Следующие k строк содержат список перемещений в формате: номер дня (от 1 до $m-1$), имя человека, название города назначения. Вы можете считать, что миллиардеры путешествуют не чаще одного раза в день, и что они отбывают поздно вечером и прибывают в город назначения рано утром следующего дня. Список упорядочен по возрастанию номера дня. Все имена и названия городов состоят не более чем из 20 латинских букв, регистр букв имеет значение. Состояния миллиардеров лежат в пределах от 1 до 100 миллиардов.

Формат выходных данных

В каждой строке должно содержаться название города и, через пробел, количество дней, в течение которых этот город лидировал по общему состоянию миллиардеров, находящихся в нём. Если таких дней не было, пропустите этот город. Города должны быть отсортированы по алфавиту (используйте обычный порядок символов: ABC...Zabc...z).

Примеры

| stdin | stdout |
|--|-----------------------------------|
| 5 Abramovich London 15000000000 Deripaska Moscow 10000000000 Potanin Moscow 50000000000 Berezovsky London 25000000000 Khodorkovsky Chita 10000000000 | Anadyr 5 London 14 Moscow 1 |
| 25 9 1 Abramovich Anadyr 5 Potanin Courchevel 10 Abramovich Moscow 11 Abramovich London 11 Deripaska StPetersburg 15 Potanin Norilsk 20 Berezovsky Tbilisi 21 Potanin StPetersburg 22 Berezovsky London | |

Задача Р. К-я строка [0.5 sec, 256 mb]

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S ;
- найти в словаре k -ю строку в лексикографическом порядке.

Изначально словарь пуст.

Формат входных данных

Первая строка входного файла содержит натуральное число N — количество команд ($N \leq 10^5$). Последующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k , либо в виде строки S , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе k -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает 10^5 .

Формат выходных данных

Для каждого числового запроса k выходной файл должен содержать k -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5 .

Примеры

| stdin | stdout |
|-----------|---------|
| 7 | tolstoy |
| pushkin | gogol |
| lermontov | |
| tolstoy | |
| gogol | |
| gorkiy | |
| 5 | |
| 1 | |