

Задача 1. Ошибка в расписании

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

В школе «Мангуст» для составления расписания уроков начали применять искусственный интеллект. Обученный на небольшой выборке удачно составленных расписаний алгоритм генерирует расписание уроков на каждый день с учётом занятых кабинетов, количества учителей, расстояния между кабинетами и прочих аспектов. Алгоритм выдаёт расписание в виде списка времён начала каждого урока. Но, к сожалению, бездумная машина допускает слишком много ошибок в своей работе, и её всё равно приходится проверять помощнику учителя информатики Константину. Константин знает, что задача проверки расписания гораздо проще, чем задача его составления, особенно когда ты знаешь правила, которые не должны быть нарушены. А правила следующие:

1. времена должны быть отсортированы по возрастанию;
2. все уроки длятся по 45 минут;
3. никакие уроки по времени не пересекаются;
4. между подряд идущими уроками всегда должна быть перемена не менее 5 минут;
5. если уроков больше трёх, то должна быть хотя бы одна большая перемена длительностью не менее 20 минут.

Константин сегодня занят, поэтому он делегировал вам разработку программы для автоматической проверки автоматически сгенерированных расписаний. Результат должен быть готов как можно скорее!

Формат входных данных

В первой строке входного файла содержится целое число T — количество составленных расписаний, которые нужно проверить ($1 \leq T \leq 10^4$).

Следующие T блоков содержат независимые друг от друга расписания. Каждое расписание начинается с целого числа N — количества уроков ($1 \leq N \leq 15$).

В следующих N строках заданы времена начала уроков в стандартном формате HH:MM.

Гарантируется, что самое раннее время для начала урока — 08:00, а самое позднее — 21:00.

Также гарантируется, что искусственный интеллект всегда генерирует корректное время.

Формат выходных данных

В выходной файл необходимо вывести ровно T строк. Каждая строка должна содержать либо слово ERROR, если в расписании есть ошибка, либо слово OK, если ошибки нет.

Система оценки

В данной задаче баллы начисляются за каждый пройденный тест отдельно. Оценка за задачу — это сумма набранных баллов за все правильно выполненные тесты.

Пример

| input.txt | output.txt |
|-----------|------------|
| 2 | ERROR |
| 3 | OK |
| 08:00 | |
| 09:30 | |
| 08:45 | |
| 1 | |
| 09:55 | |

Замечание

В первом расписании примера видно, что, во-первых, времена не упорядочены, и, во-вторых, если даже их отсортировать, то между уроками не запланирована перемена.

Задача 2. Ворд ли?

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 1 секунда |
| Ограничение по памяти: | 256 мегабайт |

Мы хотим сыграть с вами в одну игру.

Вам будет дан словарь, состоящий из пятибуквенных слов.

Мы загадаем слово из этого словаря, и у вас будет пять попыток, чтобы его отгадать.

В качестве попыток вы должны вводить слова из словаря. В ответ мы будем сообщать:

- какие из букв содержатся в загаданном слове и находятся во введенном слове на тех же позициях;
- какие содержатся, но находятся на других позициях;
- каких букв нет в загаданном слове.

Если в предложенном вами слове были буквы, которых не оказалось в загаданном слове, то вы больше не можете использовать те слова из словаря, в которых есть хотя бы одна из таких букв.

Если во введенном слове какая-то из букв встречается больше раз, чем в загаданном слове, то она выделяется столько раз, сколько она встречается в загаданном слове, причем совпавшие позиции выделяются с более высоким приоритетом.

Например, если загадано слово `horse`, а вы ввели слово `cross`, мы вернем вам строку `01120`. Это означает, что буква `s` в загаданном слове не встречается (цифра `0` в ответе), буквы `r` и `o` встречаются, но стоят на других позициях (цифра `1` в ответе), буква `s` встречается только один раз и стоит на четвертой позиции (цифра `2` в ответе). Буква `s` на пятой позиции выделяется цифрой `0`, так как она встречается в слове только один раз. Букву `s` вы больше не можете использовать в своих словах.

Протокол взаимодействия

Это интерактивная задача, и в ней вам предстоит работать не с файловым вводом-выводом, а со специальной программой — интерактором. Взаимодействие с ней осуществляется через стандартные потоки ввода-вывода.

При старте вашей программе в стандартный поток ввода подаётся одно целое число N — размер словаря ($1 \leq N \leq 10^3$).

В следующих N строках вводятся слова из словаря, по одному слову в строке. Каждое слово состоит из пяти строчных английских букв. Все слова в словаре различны.

Ваша программа должна отправлять запросы в стандартный поток вывода.

Запрос должен содержать одно слово из словаря, которое вы имеете право ввести по правилам игры.

В ответ на запрос приходит строка длины 5, состоящая из цифр `0`, `1`, `2`, расставленных согласно описанным выше правилам.

После того, как ваша программа получила ответ `22222`, она должна завершить работу.

Если введенное слово не содержится в словаре или содержит запрещенные после предыдущих попыток буквы, ваше решение получает вердикт `Wrong Answer`.

Если после пятого запроса получен ответ, отличный от `22222`, решение получает вердикт `Wrong Answer`.

Убедитесь, что вы выводите символ перевода строки и очищаете буфер потока вывода (команда `flush` языка) после каждого выведенного запроса. Иначе решение может получить вердикт `Timeout`.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 | - |
| cross | - |
| horse | - |
| stock | - |
| roast | - |
| - | cross |
| 01120 | - |
| - | roast |
| 12020 | - |
| - | horse |
| 22222 | |

Система оценки

В тестовом наборе 26 тестов. Каждый тест, кроме первого теста из условия, оценивается по отдельности максимум в 4 балла.

Если слово угадано с первой или второй попытки — решение получает 4 балла, с третьей попытки — 3 балла, с четвертой попытки — 2 балла, с пятой попытки — 1 балл.

Иначе, решение получает 0 баллов.

Замечание

Символы ‘-’ отсутствуют в вводе-выводе и используются здесь только для удобства просмотра примеров.

Задача 3. Простые сложения

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

Вот и подходит к концу занятие в математическом кружке «Мудрый филин». Занятие было посвящено свойствам делимости, после чего преподаватель предложил маленькому Иннокентию следующую задачу.

На доске написан набор чисел. У мальчика есть два варианта изменить последовательность на каждом шаге:

- стереть первые два числа, если они либо взаимно просты, либо одно из них делится на второе без остатка;
- прибавить некоторое целое число, большее 1, к первому и дописать его в конец набора. Формально, если на доске записана последовательность a_1, a_2, \dots, a_n , он может прибавить к a_1 число x и записать его в конец последовательности, получив последовательность $a_1 + x, a_2, \dots, a_n, x$.

Задача Иннокентия — за наименьшее количество операций стереть все числа. Помогите ему это сделать!

Формат входных данных

В первой строке входного файла содержится число N — количество чисел, записанных на доске ($1 \leq N \leq 10^6$).

Во второй строке расположено N целых чисел: A_i ($40 \leq A_i \leq 10^6$).

Формат выходных данных

В выходной файл необходимо вывести одно число — минимальное количество операций, необходимых для того, чтобы стереть все числа с доски.

Гарантируется, что при данных ограничениях можно стереть все числа.

Система оценки

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

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|---|---------------------------|
| 1 | 0 | Тесты из условия | |
| 2 | 2 | $N = 1, A_i \leq 10^6$ | 1 |
| 3 | 4 | $N = 2, A_i \leq 10^6$ | 1, 2 |
| 4 | 5 | $N = 3, A_i \leq 10^6$ | 1, 2, 3 |
| 5 | 7 | $N \leq 10, A_i \leq 100$ | 1 |
| 6 | 12 | $N \leq 100, A_i \leq 100$ | 1, 5 |
| 7 | 12 | $N \leq 10^6, A_i \leq 1000$ | 1, 5, 6 |
| 8 | 18 | $N \leq 10^6, A_i \leq 10^6$, все A_i — нечетные | 1 |
| 9 | 18 | $N \leq 10^6, A_i \leq 10^6$, все A_i — четные | 1 |
| 10 | 22 | $N \leq 10^6, A_i \leq 10^6$ | 1, 2, 3, 4, 5, 6, 7, 8, 9 |

Примеры

| input.txt | output.txt |
|------------|------------|
| 1 45 | 2 |
| 2 41 42 | 1 |

Замечание

Числа называются взаимно простыми, если наибольшее целое положительное число, которое делит эти числа без остатка, равно 1.

В первом примере можно к 45 прибавить 2 и стереть пару 47 и 2. Во втором примере числа взаимно простые, а значит их можно сразу стереть.

Задача 4. Шахты

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

Василию приснился страшный сон, в котором он управлял золотодобывающей шахтой. Шахта представляет собой многоуровневую систему комнат, ведущую в глубь земли. На поверхности расположено главное отделение шахты, которое, в том числе, является входом в шахту.

Отделения шахты связаны между собой тоннелями, по которым можно добраться из любого отделения шахты до любого другого, причём единственным способом. Из каждого отделения вниз ведёт не менее одного тоннеля, причём такой тоннель может вести либо в другое отделение, либо к озеру лавы. Если тоннель из отделения ведёт к озеру лавы, то эта часть шахты считается опасной и больше никаких тоннелей из отделения не было раскопано.

В каждом отделении шахты осталось некоторое количество золота, которое еще не добыли.

Василий во сне провёл исследование шахты на предмет безопасности и обнаружил, что из недр земли поднимается лава объёма L , которая может погубить всю его шахту. Любое количество лавы в отделении уничтожает всю руду этого отделения. Подъём лавы начинается по самым нижним тоннелям, распределяясь по ним равномерно, затем происходит заполнение отделения, в которое данный тоннель ведёт. Объёмы тоннелей настолько малы по сравнению с объемами отделений, что ими можно пренебречь. Если в момент заполнения отделения есть тоннель (даже закрытый), ведущий в отделение на уровень ниже, то лава начинает стекать по тоннелю в данное отделение. После переполнения какого-либо отделения лава продолжает свой путь на следующий к поверхности уровень шахты по доступным тоннелям.

Недолго думая, Василий решил закрыть доступ лаве в некоторые, возможно, все тоннели шахты. По закрытому тоннелю лава не сможет двигаться на уровни выше и окажется в нём заблокированной. Но все отделения ниже, к которым можно было попасть через этот тоннель, станут недоступны.

Существует еще одна опасность. Если перекрыть тоннели так, что вся лава не поместится в перекрытой части, то вся шахта обрушится.

Василий хочет потерять золота, как можно меньше, поэтому просит вас определить, какой минимальный объём золота будет потерян, как бы он ни закрывал тоннели.

Формат входных данных

Первая строка входного файла содержит три натуральных числа N , L , r — количество отделений шахты, объём лавы, которая поднимается из недр земли, и номер отделения, в которое можно попасть с поверхности ($1 \leq N \leq 10^3$, $1 \leq L \leq 10^4$, $1 \leq r \leq N$).

Далее следует N блоков с описаниями соответствующих отделений. Первая строка содержит три целых числа m_i , w_i , v_i — количество ведущих вниз от i -го отделения тоннелей, количество золота в i -м отделении, а также объём i -го отделения ($1 \leq i \leq N$, $0 \leq m_i < N$, $1 \leq w_i \leq 10^9$, $1 \leq v_i \leq 10^4$). Значение $m_i = 0$ означает, что из этого отделения ведёт только один тоннель в озеро с лавой.

На следующей строке находится m_i чисел — номера отделений c_{ij} , куда ведут пути от i -го отделения ($1 \leq j \leq m_i$, $1 \leq c_{ij} \leq N$).

Формат выходных данных

В выходной файл требуется вывести одно число — минимальное количество золота, которое будет потеряно.

Если объём лавы превышает суммарный объём шахты, требуется вывести -1 .

Система оценки

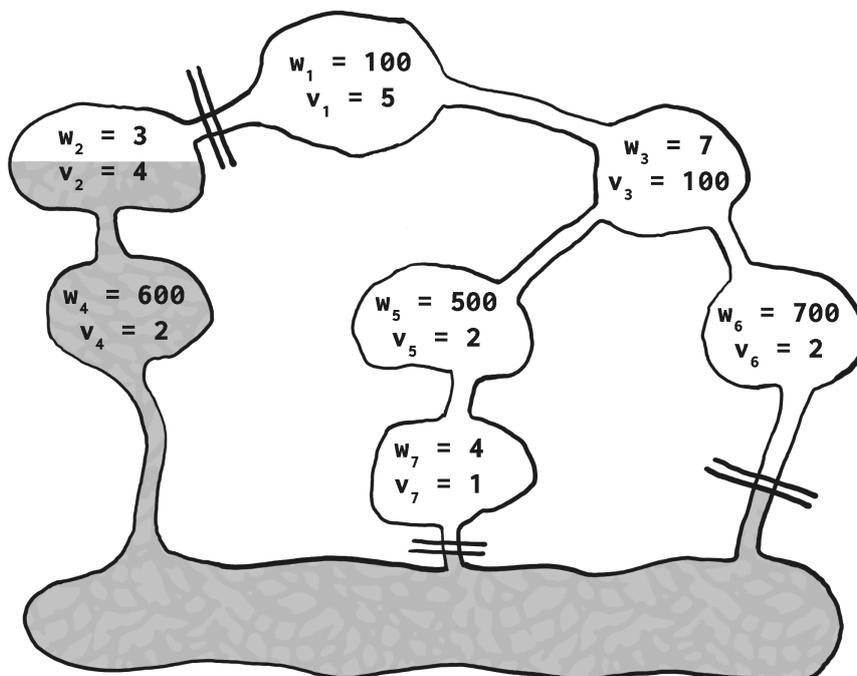
Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|---|-----------------------|
| 1 | 0 | Тесты из условия | |
| 2 | 30 | $N \cdot L \leq 10^5$ | 1 |
| 3 | 30 | Из входного отделения ведут тоннели в каждое другое отделение | 1 |
| 4 | 40 | $N \leq 10^3, L \leq 10^4$ | 1, 2, 3 |

Примеры

| input.txt | output.txt |
|--|------------|
| <pre>7 4 1 2 100 5 2 3 1 3 4 4 2 7 100 5 6 0 600 2 1 500 2 7 0 700 2 0 4 1</pre> | 603 |
| <pre>1 20 1 0 100 20</pre> | 100 |

Замечание



Изображение лучшего варианта для первого примера, в котором теряется 603 условных единиц золота.

Задача 5. Торт

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Ваня — молодой и амбициозный ученый. Однажды его статью процитировали очень много раз. По этому поводу родители купили ему торт, но не обычный: горизонтальная проекция этого торта на плоскость — выпуклый многоугольник, вместо всем привычной круглой формы.

Чтобы съесть торт, его необходимо разрезать. К этой задаче Ваня подошел с энтузиазмом — он решил поделить торт на два куска одним прямым разрезом. Разрез можно представить как прямую линию на плоскости, которая пересекает многоугольник, тем самым делит его на две части.

Ване, как и подобает ученому, стало интересно — какой получилась бы площадь большей из двух частей при определенном разрезе? Помогите ему ответить на этот вопрос.

Формат входных данных

В первой строке входного файла записано целое число N — количество вершин многоугольника ($3 \leq N \leq 10^5$).

В следующих N строках записаны по два целых числа x_i и y_i — координаты вершин ($|x_i|, |y_i| \leq 10^{14}$). Вершины многоугольника задаются в порядке обхода их по контуру по часовой стрелке.

В следующей строке записано число Q — количество прямых, задающих разрез ($1 \leq Q \leq 10^5$).

В каждой из следующих Q строк через пробел записано по четыре целых числа — координаты двух точек, задающих прямую: x_1, y_1 — первая точка, x_2, y_2 — вторая точка, соответственно ($|x_1|, |y_1|, |x_2|, |y_2| \leq 10^{14}$).

Гарантируется, что точки не совпадают.

Гарантируется, что разрез торта по каждой заданной прямой даст части торта с ненулевой площадью.

Формат выходных данных

В выходной файл необходимо вывести Q строк, в каждой из которых записано одно вещественное число — площадь наибольшей из двух получившихся частей торта. Каждый разрез считается независимо от других. Абсолютная или относительная погрешность ответа по модулю не должна превышать 10^{-6} .

Система оценки

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

| Подзадача | Баллы | Ограничения | Необходимые подзадачи |
|-----------|-------|--|-----------------------|
| 1 | 0 | Тесты из условия | |
| 2 | 22 | $N \leq 10^3, Q \leq 10^3$ | 1 |
| 3 | 15 | $N \leq 10^5, Q \leq 10^5$, точки, задающие прямые, совпадают с вершинами многоугольника | 1 |
| 4 | 16 | $N \leq 10^5, Q \leq 10^5$, точки, задающие прямые, лежат на сторонах многоугольника | 1, 3 |
| 5 | 21 | $N \leq 10^5, Q \leq 10^5$, x_1, y_1 одинаковы для всех прямых | 1 |
| 6 | 26 | $N \leq 10^5, Q \leq 10^5$ | 1, 2, 3, 4, 5 |

Пример

| input.txt | output.txt |
|-----------|------------|
| 4 | 0.5 |
| 0 0 | 0.75 |
| 0 1 | |
| 1 1 | |
| 1 0 | |
| 2 | |
| 0 0 1 1 | |
| 0 0 1 2 | |