

Задача 1. Одинаковые остатки

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

Даны два натуральных числа a и b ($1 \leq a \leq b$). Найдите два натуральных числа x и y такие, что $2 \leq x < y$, остаток от деления a на x равен остатку от деления b на y , а y делится на x . Если таких пар несколько, то x должно быть минимально возможным, а среди всех таких пар надо выбрать пару с минимальным значением y .

Формат входных данных

В первой строке входных данных находится число a , во второй — число b ($1 \leq a \leq b \leq 10^{10}$).

Обратите внимание, что входные данные могут не уместиться в стандартный 32-х битный целый тип данных.

Формат выходных данных

Если искомая пара x, y найдена, то выведите ее в одной строке, в противном случае выведите два числа -1 .

Система оценки

Тесты к этой задаче состоят из пяти подзадач.

- Тесты 1–3. Тесты из условия, оцениваются в ноль баллов.
- Тесты 4–13. В тестах этой подзадачи $b \leq 500$. Решение оценивается в 20 баллов.
- Тесты 14–23. В тестах этой подзадачи $b \leq 100\,000$. Решение оценивается в 20 баллов.
- Тесты 24–28. В тестах этой подзадачи $b \leq 1\,000\,000$. Решение оценивается в 20 баллов.
- Тесты 29–33. В тестах этой подзадачи дополнительные ограничения отсутствуют. Решение оценивается в 40 баллов.

Баллы за каждую подзадачу ставятся только при прохождении **всех** тестов группы и всех предыдущих групп.

Для каждого теста подзадачи сообщается результат работы программы участника на этом тесте.

Примеры

стандартный ввод	стандартный вывод
3 5	2 4
2 3	-1 -1
2 11	3 9

Задача 2. Надежная сеть

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

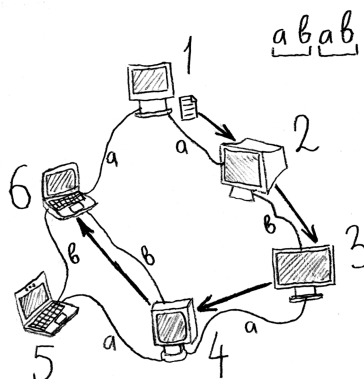
В компании «Interplanetary Software, Inc.» ведутся исследования, направленные на повышение надежности работы компьютерных сетей.

Инженеры компании построили сеть, состоящую из n компьютеров. Компьютеры соединены между собой m кабелями, по которым сигнал может ходить в обе стороны. Кабели бывают двух типов, которые обозначаются строчными латинскими буквами «a» и «b».

Компьютеры обмениваются информацией, посылая друг другу пакеты. В каждом пакете содержатся некоторые данные, а также *строка подтверждения*, изначально пустая. Каждый раз, когда пакет проходит по кабелю, то в конец строки подтверждения добавляется буква, соответствующая типу кабеля (если пакет прошел по a-кабелю, то добавляется буква «a», а если по b-кабелю — буква «b»).

Пусть два компьютера u и v обмениваются информацией. Тогда выбирается некоторый маршрут, проходящий через один или несколько компьютеров. Из-за ограничений конструкции сети маршрут не может проходить дважды через один и тот же компьютер. Затем по этому маршруту от u к v передается пакет. Получив пакет, компьютер v проверяет его строку подтверждения. Если эта строка непуста, ее длина четна и первая половина совпадает со второй, то тогда считается, что пакет доставлен надежно.

Рассмотрим рисунок ниже, на котором изображена сеть из шести компьютеров и семи кабелей.



На рисунке видно, что если выбрать маршрут $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6$ и отправить пакет по этому маршруту, то тогда строка подтверждения станет равна «abab». Ее длина четна, а правая половина («.ab») совпадает с левой («ab.»), т. е. пакет на компьютер номер 6 будет доставлен надежно.

Конечно же, построенная компанией «Interplanetary Software, Inc.» окажется бесполезной, если через нее нельзя ничего передать. Поэтому инженеры компании интересуются: существует ли такой маршрут между некоторой парой компьютеров, чтобы пакет, проходя по этому маршруту, был доставлен надежно?

Формат входных данных

В первой строке входных данных находится два целых числа n и m ($2 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$) — количество компьютеров и кабелей между ними.

В каждой из следующих m строк находится по два целых числа u_i и v_i , а также символ c_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) — компьютеры, соединенные i -м кабелем, а также тип этого кабеля.

Гарантируется, что каждая пара компьютеров соединена напрямую не более чем одним кабелем.

Формат выходных данных

Если требуемого маршрута не существует, выведите единственное целое число -1 .

Иначе в первой строке выведите целое число k ($2 \leq k \leq n$) — количество компьютеров в маршруте. Затем во второй строке выведите k целых чисел a_i ($1 \leq a_i \leq n$) — компьютеры, через которые проходит требуемый маршрут в том же порядке, в каком они следуют в маршруте. Все компьютеры в маршруте должны быть различны.

Если существует несколько способов построить требуемый маршрут, выведите любой из них.

Система оценки

№	Тесты	Дополнительные ограничения	Баллы за подзадачу
0	1 – 4	тесты из условия	0
1	5 – 31	$n \leq 200, m = n - 1, u_i = i, v_i = i + 1$	13
2	32 – 49	$n \leq 200, m = n - 1$, сеть образует дерево	17
3	50 – 67	$n \leq 2000, m = n - 1, u_i = i, v_i = i + 1$	23
4	68 – 85	$n \leq 2000, m = n - 1$, сеть образует дерево	24
5	86 – 119	Нет дополнительных ограничений	23

В подзадачах 2 и 4 дополнительное ограничение «сеть образует дерево» обозначает, что между каждой парой компьютеров существует ровно один маршрут.

Баллы за каждую из подзадач начисляются только при прохождении всех тестов этой подзадачи. Для подзадачи сообщается одно из двух:

- если все тесты пройдены, то сообщаются баллы за подзадачу,
- если хотя бы один тест не пройден, сообщается номер первого не прошедшего теста и результат работы программы участника на этом тесте.

Подзадача 5 проверяется только в случае прохождения всех остальных подзадач.

Примеры

input.txt	output.txt
6 7 1 2 a 2 3 b 3 4 a 4 5 a 5 6 b 6 1 a 4 6 b	5 1 2 3 4 6
4 3 1 2 a 2 3 b 3 4 a	-1
3 3 1 2 a 2 3 a 3 1 a	3 2 1 3
4 2 1 2 a 3 4 a	-1

Замечание

Первый пример разобран в условии задачи.

Во втором примере мы никак не можем построить требуемый маршрут, поэтому необходимо вывести -1 .

В третьем примере подходит любой маршрут длины два (например, $2 \rightarrow 1 \rightarrow 3$).

В четвертом примере невозможно построить непустой маршрут четной длины, поскольку максимально возможная длина маршрута равна 1. Обратите также внимание, что в этом примере не существует маршрута между компьютерами 1 и 3, т. е. сеть необязательно должна быть связной.

Задача 3. Добыча меди

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

Кидония — область на Марсе, которая славится своими богатыми залежами медной руды. В этой области находится n городов, выстроенных вдоль одной прямой дороги, идущей с запада на восток. Города пронумерованы целыми числами от 1 до n в порядке следования вдоль дороги. Около каждого города располагаются рудники, около i -го города находится a_i рудников.

Недавно марсиане заключили крупный контракт на поставку меди на Венеру, поэтому им требуется увеличить эффективность добычи этого металла. Но прежде всего необходимо решить проблему управления рудниками Кидонии и разбить все города на районы.

Правительство решило произвести разбиение на районы следующим образом. Сначала все города помещаются в один большой район. Далее происходит следующее:

- либо район считается конечным, т. е. не разбивается на более мелкие районы (такой район называется *терминальным*);
- либо район разбивается на два непустых района (причем города в одном районе не обязаны идти последовательно). Полученные районы могут быть рекурсивно разбиты на более мелкие по аналогичной процедуре.

После разбиения на районы можно посчитать их эффективность следующим образом:

- Эффективность терминального района равна среднему количеству рудников во всех городах этого района. Иными словами, если в районе k городов с количеством рудников $b_1, b_2, \dots, b_k$, то эффективность района равна $\frac{b_1 + b_2 + \dots + b_k}{k}$.
- Если же район не является терминальным, то его эффективность равна средней эффективности двух районов, на которые он разбивается, т. е. если он разбивается на районы с эффективностью x и y соответственно, то тогда эффективность данного района будет равна $\frac{x+y}{2}$.

Чтобы добыча меди была максимально эффективной, руководство Кидонии хочет максимизировать эффективность самого большого района (содержащего все города).

Поскольку проект по повышению эффективности пока что является экспериментальным, власти решили разбивать на районы не все города Кидонии, а только те, которые имеют номера с l по r включительно. То есть, самый большой район будет содержать только города с номерами l по r .

Какой именно отрезок городов использовать для экспериментов, правительство Кидонии пока не решило. Было утверждено q возможных вариантов. А Вам было поручено для каждого из q вариантов рассчитать максимально возможную эффективность при оптимальном делении на районы. Справитесь?

Формат входных данных

В первой строке входных данных находится целое число n ($1 \leq n \leq 10^5$) — количество городов в Кидонии.

Во второй строке входных данных находится n целых чисел a_i ($1 \leq a_i \leq 10^6$) — количество рудников в i -м городе.

В третьей строке входных данных расположено целое число q ($1 \leq q \leq 10^5$) — количество вариантов, которые необходимо рассмотреть.

В каждой из следующих q строк находится по два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$). Это означает, что в i -м варианте требуется разбивать на районы лишь те города, которые имеют номера с l_i по r_i включительно.

Формат выходных данных

Выведите q строк. В i -й строке выведите вещественное число e_i — максимально возможную эффективность разбиения на районы всех городов с номерами от l_i до r_i .

Выводить ответ следует с абсолютной погрешностью не более 10^{-4} . Иными словами, если ответ жюри для i -го варианта равен o_i , а Ваш ответ равен e_i , то он считается правильным, если $|o_i - e_i| \leq 10^{-4}$.

Система оценки

№	Тесты	Дополнительные ограничения	Баллы за подзадачу
0	1	тест из условия	0
1	2 – 31	$n \leq 15, q = 1$	18
2	32 – 41	$n \leq 3000, q = 1$	9
3	42 – 61	$q = 1$	19
4	62 – 125	$a_i \leq 3$	26
5	126 – 189	Нет дополнительных ограничений	28

Баллы за каждую из подзадач начисляются только при прохождении всех тестов этой подзадачи.

Для подзадачи сообщается одно из двух:

- если все тесты пройдены, то сообщаются баллы за подзадачу,
- если хотя бы один тест не пройден, сообщается номер первого не прошедшего теста и результат работы программы участника на этом тесте.

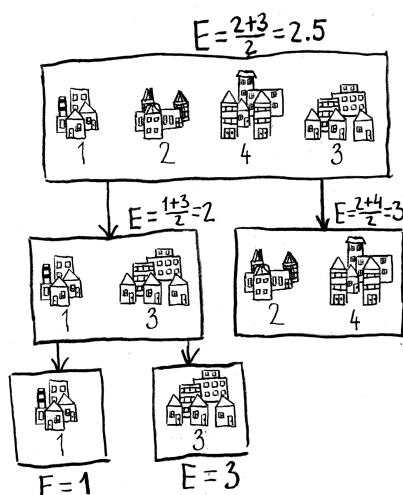
Подзадача 5 проверяется только в случае прохождения всех остальных подзадач.

Пример

input.txt	output.txt
7	3.1250000000
2 1 2 4 3 3 3	3.0000000000
3	1.7500000000
2 5	
5 7	
1 3	

Замечание

В первом запросе требуется разбить на районы города $\{1, 2, 4, 3\}$. Одно из разбиений может выглядеть следующим образом:



Как видно из рисунка, эффективность такого разбиения равна 2.5. Однако оптимальным является разбиение с эффективностью 3.125.

Во втором запросе необходимо разбить на районы города $\{3, 3, 3\}$. Здесь оптимальным является любое разбиение, в частности, можно вообще не проводить разбиение на более мелкие районы, и эффективность такого разбиения будет равна $\frac{3+3+3}{3} = 3$.

В третьем запросе можно разбить города на два района, $\{1, 2\}$ (эффективность 1.5) и $\{2\}$ (эффективность 2). Тогда эффективность всех городов будет равна $\frac{1.5+2}{2} = 1.75$.

Задача 4. Города на Венере

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

На планете Венера обитает немногочисленная, но очень развитая космическая цивилизация. Жители этой планеты построили n городов, i -й город имеет название a_i . У немногочисленной цивилизации и алфавит небольшой: жители Венеры используют только три буквы: «a», «b» и «c».

Два жителя Венеры (будем для простоты называть их Артем и Борис) решили сыграть в города. Правила игры такие. Сначала Артем называет некоторый город b_0 . Затем Борис должен назвать какой-либо город b_1 , который начинается на ту же букву, на которую заканчивается b_0 . Далее Артем называет город b_2 , который начинается на ту же букву, на которую заканчивается предыдущий город. Далее игра продолжается аналогичным образом, причем названные города не должны повторяться. Проигрывает тот, кто не может сделать ход.

Артем и Борис часто путешествуют вместе и во время долгих поездок часто играют в города. Настолько часто, что игра им надоела, и они стали искать оптимальную стратегию. Потратив на поиск стратегии не один день во время длительных поездок, друзья так ни к чему и не пришли. Поэтому они попросили помощи у Вас.

Вам требуется определить, кто выигрывает в города, если игроки действуют оптимально. Учитывайте, что Артем и Борис — опытные путешественники, поэтому знают названия всех n городов на Венере наизусть.



Формат входных данных

В первой строке входных данных находится целое число n ($1 \leq n \leq 10^5$) — количество городов на Венере.

В каждой из следующих n строк находится строка a_i , состоящая из строчных латинских букв «a», «b», «c» — название i -го города. Никакие два города не имеют одинаковое название.

Гарантируется, что суммарная длина всех названий городов не превосходит $2 \cdot 10^6$ символов.

Формат выходных данных

Если при оптимальной игре побеждает Артем, выведите «Artem» (без кавычек). Иначе выведите «Boris» (без кавычек).

Система оценки

№	Тесты	Дополнительные ограничения	Баллы за подзадачу
0	1 — 3	тесты из условия	0
1	4 — 40	$n \leq 10$	7
2	41 — 64	$n \leq 17$	18
3	65 — 107	$n \leq 40$	19
4	108 — 137	$n \leq 300$	16
5	138 — 236	Названия всех городов не содержат букву «c»	26
6	237 — 253	Нет дополнительных ограничений	14

Баллы за каждую из подзадач начисляются только при прохождении всех тестов этой подзадачи. Для подзадачи сообщается одно из двух:

- если все тесты пройдены, то сообщаются баллы за подзадачу,
- если хотя бы один тест не пройден, сообщается номер первого не прошедшего теста и результат работы программы участника на этом тесте.

Подзадача 6 проверяется только в случае прохождения всех остальных подзадач.

Примеры

input.txt	output.txt
3 abacaba abaab acc	Artem
2 a aa	Boris
3 abbaab babbaa baba	Artem

Замечание

В первом примере Артем может назвать, например, город «acc». Так как городов, начинающихся на «с», нет, то Борис не может сделать ход и проигрывает.

Во втором примере, если Артем назовет «aa», Борис назовет «a», и наоборот. После этого городов не останется, и Борис победит.

В третьем примере Артем называет «baba». Затем Борис вынужден ответить «abbaab». Далее Артем отвечает «babbaa» и выигрывает, поскольку города закончились, и Борис не сможет сделать ход.