

Задача А. ПОИ

Имя входного файла:	poi.in
Имя выходного файла:	poi.out
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Питерская Олимпиада по Информатике (ПОИ) проходила согласно таким необычным правилам. Было n участников и t задач. Каждая задача оценивалась с использованием всего лишь одного теста. Таким образом, для каждого участника и каждой задачи было только две возможности: либо участник решил задачу, либо не решил ее. Не было частичного оценивания решения никакой задачи.

Количество баллов, назначенное каждой задаче, определялось после соревнования, и было равно количеству участников, которые не решили задачу. Баллы каждого участника подсчитывались как сумма баллов, назначенных задачам, решенным этим участником.

Перед соревнованием участникам присвоили уникальные номера от 1 до n включительно. Филиппу был присвоен номер p . В финальном протоколе участники перечислены в порядке убывания набранных ими баллов. В случае равенства баллов, первыми будут перечислены участники, которые решили больше задач. В случае равенства количества решенных задач, участники с одинаковыми результатами будут перечислены в порядке возрастания их номеров.

Филипп участвовал в соревновании, но он запутался в сложных правилах оценивания, и сейчас он, глядя на результаты, не в состоянии определить свое место в финальном протоколе. Помогите Филиппу написать программу, которая подсчитает его баллы и место в финальном протоколе.

Формат входных данных

Первая строка входных данных содержит целые числа n , t и p — количество участников, количество задач и номер, присвоенный Филиппу ($1 \leq p \leq n \leq 2000$, $1 \leq t \leq 2000$).

Последующие n строк описывают, какие задачи были решены какими участниками. k -я строка из них описывает, какие задачи были решены участником с номером k . Каждая такая строка содержит t целых чисел, разделенных пробелом. Первое из этих чисел обозначает, решил ли первую задачу участник с номером k . Второе число обозначает то же для второй задачи, и так далее. Эти t чисел могут быть только 0 или 1, где 1 означает, что участник с номером k решил соответствующую задачу, а 0 означает, что он ее не решил.

Формат выходных данных

Выведите два целых числа, разделенных одним пробелом. Первое число — количество баллов, полученных Филиппом на ПОИ. Второе число — место Филиппа в финальном протоколе. Место — это целое число от 1 до n , где 1 означает, что участник расположен сверху финального протокола (то есть имеет наибольшее количество баллов), а n означает, что он расположен внизу финального протокола (то есть имеет наименьшее количество баллов).

Система оценки и описание подзадач

Для набора тестов общей стоимостью 35 баллов, никто из участников не наберет столько же баллов, как Филипп.

Пример

poi.in	poi.out
5 3 2	3 2
0 0 1	
1 1 0	
1 0 0	
1 1 0	
1 1 0	

Замечание

Первая задача не была решена одним участником, она оценивается в 1 балл. Вторая задача не была решена двумя участниками, она оценивается в 2 балла. Третья задача не была решена четырьмя участниками, она оценивается в 4 балла. Таким образом, первый участник наберет 4 балла. Второй участник (Филипп), а также четвертый и пятый участники наберут по 3 балла каждый. Третий участник наберет 1 балл. Участник с номерами 2, 4 и 5 имеют одинаковое количество баллов и решили одно и то же количество задач, поэтому в соответствии со вторым правилом (в этом случае участники располагаются в порядке возрастания их номеров), Филипп окажется перед участниками с номерами 4 и 5. Таким образом, в финальном протоколе Филипп будет на втором месте, после участника с номером 1.

Задача В. Парковка

Имя входного файла:	garage.in
Имя выходного файла:	garage.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Парковка имеет n мест, пронумерованных от 1 до n . Парковка открывается пустой каждое утро и работает на протяжении дня следующим образом. Когда автомобиль приезжает на парковку, парковщик проверяет, есть ли свободные места. Если таковых нет, автомобиль ожидает возле въезда до тех пор, пока освободится какое-то место. Если есть свободное место, или как только оно освобождается, автомобиль занимает свободное парковочное место. Если есть несколько свободных парковочных мест, автомобиль занимает место с наименьшим номером. Когда приезжают парковаться другие автомобили, но уже есть ожидающий автомобиль, они выстраиваются в очередь на въезде в том порядке, в котором приехали. После того, как освобождается парковочное место, его занимает первый автомобиль из очереди (то есть тот, который прибыл парковаться первым).

Стоимость парковки одного автомобиля в долларах определяется как произведение веса этого автомобиля в килограммах на тариф его парковочного места. Стоимость парковки автомобиля не зависит от того, сколько времени этот автомобиль находится на парковке.

Парковщик знает, что сегодня на парковку приедет m автомобилей, и он знает порядок их приезда и отъезда. Помогите ему подсчитать, сколько долларов он сегодня заработает.

Формат входных данных

Первая строка входных данных содержит два целых числа n и m — количество парковочных мест и автомобилей ($1 \leq n \leq 100$, $1 \leq m \leq 2000$).

Следующие n строк описывают тарифы парковочных мест, s -я из этих строк содержит одно целое число r_s — тариф парковочного места с номером s в долларах за килограмм ($1 \leq r_s \leq 100$).

Следующие m строк описывают веса автомобилей. Автомобили пронумерованы в произвольном порядке от 1 до m включительно, k -я из этих m строк содержит одно целое число w_k — вес автомобиля с номером k в килограммах ($1 \leq w_k \leq 10000$).

Следующие $2m$ строк описывают приезд и выезд всех автомобилей в хронологическом порядке. Положительное целое число i показывает, что автомобиль с номером i приезжает на парковку. Отрицательное целое число $-i$ показывает, что автомобиль с номером i уезжает с парковки.

Гарантируется, что никакой автомобиль не выезжает с парковки до своего приезда, и все автомобили от 1 до m включительно появятся в этой последовательности строк ровно 2 раза, один раз как приезжающий, второй — как выезжающий. Также гарантируется, что никакой из автомобилей не выедет с парковки, пока не займет место на парковке (то есть, никакой автомобиль не уедет пока стоит в очереди).

Формат выходных данных

Выведите одно целое число — количество долларов, которое заработает парковщик.

Система оценки и описание подзадач

Для набора тестов общей стоимостью 40 баллов всегда будет хотя бы одно свободное парковочное место для каждого приезжающего автомобиля. В этих тестах никакой автомобиль не должен будет ожидать освобождения парковочного места.

Примеры

garage.in	garage.out
3 4 2 3 5 200 100 300 800 3 2 -3 1 4 -4 -2 -1	5300
2 4 5 2 100 500 1000 2000 3 1 2 4 -1 -3 -2 -4	16200

Задача С. Расшифровка письменности Майя

Имя входного файла: `writing.in`
Имя выходного файла: `writing.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Расшифровка письменности Майя оказалась более сложной задачей, чем предполагалось ранними исследованиями. На протяжении более чем двух сотен лет удалось узнать не так уж много. Основные результаты были получены за последние 30 лет.

Письменность Майя основывается на маленьких рисунках, известных как значки, которые обозначают звуки. Слова языка Майя обычно записываются с помощью этих значков, которые располагаются рядом друг с другом в некотором порядке. Одна из проблем расшифровки письменности Майя заключается в определении этого порядка. Рисуя значки некоторого слова, писатели Майя иногда выбирали позиции для значков, исходя скорее из эстетических взглядов, а не определенных правил. Это привело к тому, что, хотя звуки для многих значков известны, археологи не всегда уверены, как должно произноситься записанное слово.

Археологи ищут некоторое слово w . Они знают значки для него, но не знают все возможные способы их расположения. Они просят Вас о помощи. Они дадут Вам g значков, составляющих слово w , и последовательность s всех значков в надписи, которую они изучают, в порядке их появления. Помогите им, подсчитав количество возможных появлений слова w .

Напишите программу, которая по значкам слова w и по последовательности s значков надписи подсчитывает количество всех возможных вхождений слова w в s ; то есть количество всех различных позиций идущих подряд g значков в последовательности s , которые являются какой-либо перестановкой значков слова w .

Формат входных данных

Первая строка входных данных содержит два целых числа g и $|s|$ — количество значков в слове w и количество значков в последовательности s ($1 \leq g \leq 3000$, $g \leq |s| \leq 3 \cdot 10^6$).

Формат выходных данных

Выведите единственное целое число — количество возможных вхождений слова w в s .

Система оценки и описание подзадач

Для части тестов, оцениваемых в 50 баллов, выполняется ограничение $g \leq 10$.

Пример

<code>writing.in</code>	<code>writing.out</code>
4 11 cAda AbrAcadAbRa	2

Задача D. Пробки на дорогах

Имя входного файла: `traffic.in`
Имя выходного файла: `traffic.out`
Ограничение по времени: 6 секунд
Ограничение по памяти: 256 мегабайт

Берляндия состоит из n городов. Каждому городу присвоен уникальный целочисленный номер от 0 до $n - 1$. Некоторые пары городов соединены дорогами. Все дороги двусторонние, и между любыми двумя городами существует ровно один маршрут. Маршрут, соединяющий города c_0 и c_k — это последовательность различных городов c_0, \dots, c_k такая, что существует дорога из c_{i-1} в c_i для всех $1 \leq i \leq k$.

В рамках подготовки к чемпионату мира по керлингу 3018 года в одном из городов Берляндии должен быть построен новый стадион, на котором будет проведена главная игра всего чемпионата. После игры все фанаты, кроме тех, что живут в городе со стадионом, едут из города со стадионом в свой родной город. Величина пробки на каждой дороге пропорциональна количеству фанатов, которые проедут по этой дороге. Вам необходимо выбрать такое местонахождение стадиона, чтобы величина пробки на дороге с самой большой пробкой была минимально возможной. Если существуют несколько городов, одинаково подходящих для расположения стадиона, вы можете выбирать любой из них.

Формат входных данных

Первая строка входных данных подержит единственное целое число n — количество городов в Берляндии ($1 \leq n \leq 10^6$).

Следующая строка содержит n целых чисел p_i — количество фанатов, которые живут в i -м городе ($1 \leq p_i \leq 2 \cdot 10^9$). Гарантируется, что общее количество фанатов, живущих во всех городах Берляндии, не превосходит $2 \cdot 10^9$.

Каждая из следующих $n - 1$ строк содержат два числа u_i и v_i — пары городов, соединенных дорогами ($0 \leq u_i, v_i, \leq n - 1$).

Формат выходных данных

Выведите одно целое число — номер города, в котором необходимо построить стадион.

Система оценки и описание подзадач

Эта задача состоит из четырех подзадач.

Подзадача 1 (25 баллов)

Предположим, что все города лежат на прямой от востока к западу, и все дороги идут по прямой без ветвлений. Более формально, для всех i из диапазона $0 \leq i \leq n - 2$ выполняются соотношения $u_i = i$ и $v_i = i + 1$. Количество городов не превосходит 1000.

Подзадача 2 (25 баллов)

Выполняется предположение из подзадачи 1, но количество городов может достигать 10^6 .

Подзадача 3 (25 баллов)

Предположение из подзадачи 1 более не выполняется. Количество городов не превосходит 1000.

Подзадача 4 (25 баллов)

Предположение из подзадачи 1 не обязательно выполняется. Количество городов не превосходит 10^6 .

Пример

<code>traffic.in</code>	<code>traffic.out</code>
5	3
10 10 10 20 20	
0 2	
1 2	
2 3	
3 4	

Задача E. Ячейки

Имя входного файла: `boxes.in`
Имя выходного файла: `boxes.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано $2n$ ячеек, расположенных в ряд ($n \leq 5$). Две соседние ячейки пусты, остальные содержат $n - 1$ символ «A» и $n - 1$ символ «B». Пример для $n = 5$: `|A|B|B|A|.|. |A|B|A|B|`

Правило обмена: содержимое двух соседних непустых ячеек может быть перемещено в пустые ячейки, сохраняя порядок.

Определите, какое минимальное количество операций обмена необходимо сделать, чтобы все символы «A» располагались левее всех символов «B». При этом не имеет значения, какие ячейки останутся пустыми.

Формат входных данных

Первая строка входных данных содержит единственное число n ($2 \leq n \leq 5$).

Вторая строка содержит $2n$ символов, описывающих состояние ячеек. Символ «.» обозначает пустую ячейку. Строка содержит ровно $n - 1$ символ «A», ровно $n - 1$ символ «B», а также какие-то два соседних символа равны «.».

Формат выходных данных

Если нельзя добиться того, чтобы все «A» располагались левее всех «B», выведите единственное число -1 .

В противном случае в первой строке выведите минимальное количество операций m , за которое можно этого добиться. В следующих $m + 1$ строке выведите все промежуточные состояния ячеек, включая начальное и конечное. Для лучшего понимания формата выходных данных изучите тест из примера.

Примеры

<code>boxes.in</code>	<code>boxes.out</code>
5 ABBA..ABAB	3 ABBA..ABAB ABVABA..B A..ABAABBB AAAAB..BBB