

Задача А. Письма долгого хранения

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Бомбослава — очень занятая девушка. Она вовлечена в множество проектов, поэтому важные письма, требующие ответа, переполняют её почтовый ящик в огромных количествах. У Бомбославы очень редко бывает время сразу ответить на письмо, поэтому она просто держит каждое письмо непрочитанным до тех пор, пока её не начнёт мучить совесть. Когда это случается, она не находит другого выхода, кроме как начать отвечать на письма. Это происходит, если выполняется хотя бы одно из двух условий:

- количество непрочитанных писем становится равным a ;
- самое старое непрочитанное письмо поступило b часов назад.

К сожалению, Бомбославе быстро надоедает отвечать на письма, поэтому в середине процесса она останавливается. А именно, если она начинает разбирать почту, когда в ней имеется k непрочитанных писем, то она прочтает и ответит ровно на $\lceil \frac{k}{2} \rceil$ (половину от k , округлённую вверх) наиболее старых из них.

Известно, что всего Бомбославе придёт n писем, i -е из которых появится в начале t_i -го часа, причём больше одного письма в час ей не приходит. Каждый очередной час Бомбослава начинает с проверки, не пришли ли ей новые письма в начале этого часа, после чего она принимает решение, пора ли уже отвечать на скопившиеся письма, или можно ещё подождать.

Формат входных данных

В первой строке входных данных записаны три целых числа n , a и b ($1 \leq n, a \leq 100$, $1 \leq b \leq 1000$) — количество писем, которые последовательно поступают на почту Бомбославе, минимальное количество непрочитанных сообщений, уже **не являющееся** допустимым, и минимальное количество часов не прочтения письма, также уже **не являющееся** допустимым представлениям Бомбославы.

В следующей строке записаны в порядке возрастания n различных чисел $t_1, t_2, \dots, t_n$ ($1 \leq t_1 < t_2 < \dots < t_n \leq 1000$) — номера часов, в которые соответствующие письма приходят Бомбославе на почту.

Формат выходных данных

Выведите n целых чисел, i -е из которых задаёт час, в который Бомбослава прочтает i -е письмо.

Примеры

стандартный ввод	стандартный вывод
4 10 100 1 2 3 4	101 101 103 104
8 4 50 1 2 3 4 50 90 91 100	4 4 53 53 100 100 141 150

Замечание

В первом примере, в начале часа 101 Бомбославу первый раз начнёт мучать совесть. Поскольку в этот момент в её почтовом ящике окажется 4 письма, то она прочтает ровно два письма. Теперь самое раннее непрочитанное письмо поступило в начале часа 3, поэтому следующий раз Бомбославу начнёт мучать совесть в начале часа 103. Поскольку непрочитанных писем на почте будет только 2, то Бомбослава прочтает одно из них и остановится. Наконец, в начале часа 104 совесть снова потребует от Бомбославы разобраться с почтой, и она прочтёт последнее имеющееся письмо.

Во втором примере Бомбослава будет разбирать почту пять раз.

1. В начале 4 часа на почту поступит письмо 4, и их количество станет равно $a = 4$, поэтому она прочитает первые два.
2. В начале 53 часа самому старому письму будет $b = 50$ часов, поэтому она снова прочитает первые два (округлённый вверх результат от деления 3 на 2).
3. В начале часа 100 часа на почте будет только 3 письма, но самое старое из них пришло в начале часа 50. Бомбослава снова проверит два наиболее старых из них.
4. В начале 141 часа на почте будут только письма, пришедшие в момент времени 91 и 100 и Бомбослава проверит перове из них.
5. Наконец, в начале 150 часа Бомбослава будет разгребать почту последний раз и ответит на самое последнее письмо.

Задача В. Девчата против автомата

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дуся и Люся приехали на Школу Олимпиадного Программирования в городе Ретрополис. В общежитии, где их поселили, расположен автомат по продаже кофе и различных снеков. Проблема заключается в том, что автомат принимает только монеты достоинством 1, 5 и 10 бурлей, а также купюры в 50 и 100 бурлей. Купюры по 500, 1000 и 5000 бурлей автомат не принимает. Другие монеты и купюры в Берляндии не используются.

Дуся и Люся не испытывают недостатка в деньгах, однако, поскольку они постоянно пользуются данным автоматом, им регулярно не хватает монет и купюр нужного достоинства. Сегодня вечером они мило посидели в местечке «Квас&Котлеты», так что итоговый счёт составил n бурлей. Девушки решили разбить счёт на две части, каждая из которых будет не меньше a бурлей. При этом они хотят получить в качестве сдачи такой набор купюр и монет, в котором суммарный номинал пригодных для использования в автомате купюр и монет как можно больше. Оплата счёта размера x происходит по следующему алгоритму:

1. Расплачивающаяся девушка отдаёт официанту минимальное количество купюр достоинством 5000, составляющее сумму хотя бы x . Пусть это количество равно k .
2. Официант отдаёт сдачу $5000k - x$, используя минимально возможное количество купюр и монет. Можно показать, что такой набор купюр и монет всегда определяется единственным образом для любой возможной величины сдачи.

Помогите Люсе и Дусе разбить сумму n на две суммы x и $n - x$ так, чтобы каждая из них была не меньше a , и суммарный номинал всех подходящих для автомата монет и купюр в **сдаче** обоих девушек был как можно больше.

Формат входных данных

В единственной строке входных данных записаны два целых числа n и a ($1 \leq 2 \cdot a \leq n \leq 10^{18}$) — количество денег, которое требуется заплатить за приятный вечер, и минимальная сумма, которую должна внести каждая из подруг.

Формат выходных данных

Выведите одно целое число — максимальная сумма сдачи, которую можно будет использовать в вендинговом аппарате.

Примеры

стандартный ввод	стандартный вывод
1000 500	0
10000 4000	500
623 57	877

Замечание

В первом тесте из условия единственный допустимый способ разбить чек на две части — это заплатить по 500 бурлей. Сдача, полученная каждой из подруг, будет составлять $5000 - 500 = 1000 + 1000 + 1000 + 1000 + 500$ бурлей, которые использовать в вендинговом автомате не получится.

Во втором тесте из условия можно, например, разбить сумму в 10000 бурлей на части по 4200 бурлей и 5800 бурлей. Сдача с первой части составит $5000 - 4200 = 500 + 100 + 100 + 100$ бурлей, а сдача со второй части составит $10000 - 5800 = 1000 + 1000 + 1000 + 1000 + 100 + 100$, то есть, в распоряжении подруг окажется пять купюр по 100 бурлей, которые можно потратить в автомате.

Задача С. Эффективный менеджмент возвращается

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

В попытках сделать работу программистов более плодотворной и комфортной менеджерам по персоналу порой приходится решать самые необычные задачи. В штате крупной ИТ-компании «Мягкие изделия двигателей рядом» имеется n сотрудников, пронумерованных последовательными целыми числами от 1 до n . Также HR-отделу компании известно m пар сотрудников, которые недолюбливают друг друга. Менеджеры хотят выбрать какое-нибудь число k , после чего распределить всех программистов между k командами удовлетворив следующие пожеланию директора фирмы:

1. Всем должно быть комфортно работать, поэтому никакая пара программистов, недолюбливающих друг друга, не должна оказаться в одной команде.
2. Разбиение на команды должно быть эффективным, поэтому никакие две команды должно быть нельзя объединить в одну, так чтобы не нарушить пункт 1.

Обратите внимание, что параметр k можно выбрать любым подходящим, и от вас не требуется его максимизировать или минимизировать.

Формат входных данных

В первой строке записаны два числа n и m ($1 \leq n \leq 200\,000$, $0 \leq m \leq 200\,000$) — количество сотрудников компании и количество пар недолюбливающих друг друга сотрудников, о которых известно HR-отделу.

В каждой из последующих m строк записаны два числа u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$) — номера сотрудников соответствующей пары. Гарантируется, что никакая пара не встречается во входных данных дважды.

Формат выходных данных

В первой строке вывода напечатайте количество команд k . Во второй строке выведите n целых чисел $c_1, c_2, \dots, c_n$ от 1 до k включительно, означающих, что сотрудник i должен быть отнесён к команде c_i . Каждой команде от 1 до k должен быть назначен хотя бы один сотрудник, никакие два недолюбливающих друг друга сотрудника не должны быть в одной команде, но при этом никакие две команды нельзя объединить, так чтобы это условие не нарушилось.

Примеры

стандартный ввод	стандартный вывод
3 2 1 2 2 3	2 1 2 1
5 6 1 2 2 3 1 3 5 4 5 3 4 3	3 1 2 3 1 2

Задача D. Афера

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Недавно Остапу Бендеру пришёл в голову 401-й сравнительно честный способ отъёма денег у населения, а главное — совершенно безопасный! Великий комбинатор заметил, что население Старгорода крайне пристально следит за выступлениями местной команды «Звёздные псы» в профессиональной футбольной лиге вымышленных городов, а главное, обожает делать денежные ставки на исходы матчей данной команды.

Остапу уже поступило n предложений заключить пари на результат ближайшего матча. Каждый из предложивших хочет поставить на один из трёх возможных исходов матча: победа команды любимцев жителей Старгорода, их ничья или поражение. В случае, если исход матча в точности совпадёт со сделанной ставкой, то Остап будет вынужден выплатить сделавшему её гражданину a_i бурлей. При любом другом исходе сделавший ставку сам будет должен заплатить Остапу b_i бурлей.

План Остапа заключается в том, чтобы выбрать некоторые из сделанных предложений и принять соответствующие пари, отклонив все остальные под каким-нибудь предлогом. Остап хочет выбрать такое подмножество ставок, которое гарантирует ему максимальную прибыль **в худшем для него случае**.

Формат входных данных

В первой строке входных данных записано одно целое число n ($1 \leq n \leq 500$) — количество предложений о пари на результат матча, поступивших Остапу.

Каждая из последующих n строк содержит одно описание предложения о пари, состоящее из символа t_i и двух чисел a_i и b_i ($1 \leq a_i, b_i \leq 500$), где t_i это «W», «D» или «L» в зависимости от того, делается ли ставка на, соответственно, победу, ничью или поражение любимой команды Старгорода, a_i равняется количеству бурлей, которое должен будет заплатить Остап в случае наступления заявленного в ставке исхода, а b_i равняется количеству бурлей, которое заплатит Остапу делающий ставку, если исход игры будет отличаться от заявленного им.

Формат выходных данных

Выведите одно целое число равное максимальной прибыли, которую может гарантировать себе Остап, если правильно выберет, какие пари принять.

Примеры

стандартный ввод	стандартный вывод
3 W 2 2 D 2 2 L 2 2	2
3 W 3 1 D 3 1 L 3 1	0
5 W 3 3 W 2 2 D 10 100 L 3 1 D 3 3	1

Задача Е. Случайная величина моды

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Хотя в Москве ещё идёт снег, а лето, возможно, вообще в этом году не наступит, Глеб всё-таки решил подготовиться к тёплому сезону и купить себе стильный бежевый плащ. С этой целью он приехал в огромный торговый центр, состоящий из n магазинов, выстроенных в одну длинную цепочку. Магазины пронумерованы от 1 до n в порядке следования. Для всех i от 1 до $n-1$ магазины i и $i+1$ являются соседними, и перейти между ними можно ровно за одну секунду. Глеб только что припарковался около магазина с номером x и собирается обойти торговый центр в поисках подходящего плаща.

Про каждый магазин Глеб знает, что если он в него зайдёт, то проведёт в нём ровно t_i секунд, перед тем как поймёт есть ли в нём подходящий плащ или нет, при этом заранее известно, что плащ окажется в магазине с номером i с вероятностью p_i процентов. Плащ можно обнаружить в каждом магазине независимо от любого другого магазина, в частности, нужный плащ может быть как в нескольких магазинах, так и ни в одном из них. Глеб посетит каждый магазин не более одного раза, при этом если Глеб оказывается около магазина, в котором ещё не был, то **обязательно заходит** и полностью его осматривает, даже если знает, что плаща в магазине точно нет, то есть $p_i = 0$.

Поскольку Глеб не большой любитель разгуливать по торговым центрам, то, как только он найдёт подходящий плащ, то тут же купит его и пойдёт обратно к магазину с номером x . Теперь ему интересно, если он будет в каждый момент времени принимать оптимальные решения о дальнейших действиях, чему равно математическое ожидание времени, которое он проведёт в торговом центре? Обратите внимание, Глеб уезжает из торгового центра только когда находит нужный плащ, или не остаётся ни одного ещё не посещённого магазина.

Формат входных данных

В первой строке записаны два числа n и x ($1 \leq x \leq n \leq 300\,000$) — количество магазинов в торговом центре, и номер магазина, напротив которого припарковался Глеб.

В i -й из следующих n строк записаны два числа p_i и t_i ($0 \leq p_i \leq 100$, $1 \leq t_i \leq 1000$) — вероятность, что в i -м магазине найдётся искомый Глебом плащ, выраженная целым числом процентов, и количество секунд, которые Глеб потратит на поиски плаща в данном магазине.

Формат выходных данных

Выведите одно вещественное число — математическое ожидание количества секунд, через которое Глеб покинет торговый центр при оптимальной стратегии посещения магазинов. Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не будет превосходить 10^{-6} .

А именно: пусть ваш ответ равен a , а ответ жюри — b . Проверяющая программа будет считать ваш ответ правильным, если $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$.

Примеры

стандартный ввод	стандартный вывод
5 1 50 1 50 1 50 1 50 1 50 1	3.8125000000
4 2 1 1 0 10 0 10 100 1000	1016.8600000000

Задача F. Семь раз отмерь, один раз раздели

Имя входного файла: стандартный ввод или `input.txt`
Имя выходного файла: стандартный вывод или `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Бомбославу очень нравится решать задачи по программированию с использованием принципа «Разделяй и властвуй», однако его не покидает ощущение, что для некоторых деревьев этот принцип можно применять более эффективно, чем это достигается при использовании центроидной декомпозиции (*для решения задачи знание этого понятия необязательно*).

Бомбослав хочет по произвольному дереву определять минимально возможную глубину рекурсии при использовании стратегии «Разделяй и властвуй». Опишем процедуру формально. Пусть дано дерево, состоящее из n вершин. Пока в графе имеется хотя бы одна вершина, повторяется следующая процедура:

1. в каждой компоненте связности выбирается ровно одна вершина;
2. выбранные вершины удаляются, как и все инцидентные им рёбра.

За какое минимальное количество шагов вышеописанной процедуры можно удалить все вершины изначального дерева?

Формат входных данных

В первой строке входных данных записано число n ($1 \leq n \leq 300\,000$) — количество вершин в имеющемся у Бомбослава дереве.

В i -й из последующих $n - 1$ строк записаны два числа u_i и v_i ($1 \leq u_i, v_i \leq n$) — номера вершин, соединённых i -м ребром.

Формат выходных данных

Выведите одно целое число — минимальное количество шагов, за которое Бомбослав сможет удалить все вершины из дерева.

Примеры

стандартный ввод	стандартный вывод
3 2 3 1 3	2
11 1 2 2 3 3 4 3 5 3 6 4 7 4 8 4 9 4 10 4 11	3