

Задача А. Автобусы

Имя входного файла: `ain.txt`
Имя выходного файла: `aout.txt`
Максимальное время работы на одном тесте: 1 секунда
Максимальный объем используемой памяти: 64 мегабайта

Для заезда в оздоровительный лагерь организаторы решили заказать автобусы. Известно, что в лагерь собираются поехать N детей и M взрослых. Каждый автобус вмещает K человек. В каждом автобусе, в котором поедут дети, должно быть не менее двух взрослых.

Определите, удастся ли отправить в лагерь всех детей и взрослых, и если да, то какое минимальное количество автобусов требуется для этого заказать.

Формат входных данных

Во входном файле записаны 3 натуральных числа через пробел – N , M и K , каждое не превосходит 10 000.

Формат выходных данных

В выходной файл выведите количество автобусов, которые нужно заказать. Если же отправить всех в лагерь невозможно, выведите 0 (ноль).

Примеры

<code>ain.txt</code>	<code>aout.txt</code>
10 4 7	2
10 4 5	0

Задача В. Лавочки

Имя входного файла:	bin.txt
Имя выходного файла:	bout.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта

Лавочки в парке устроены следующим образом. Несколько одинаковых кубических гранитных блоков ставятся в ряд, а на них кладется гранитная плита (см. рисунок). Архитектор-модернист решил, что будет интереснее, если у всех лавочек расположение гранитных блоков-ножек будет разным (и не обязательно симметричным). При этом они располагаются так, чтобы плита не падала: для этого достаточно, чтобы и слева, и справа от центра плиты был хотя бы один гранитный блок или его часть (в частности, если центр плиты приходится на середину какого-нибудь блока, то и слева, и справа от центра плиты находится часть блока, и плита не падает).



Грабители обнаружили, что можно по одному вытаскивать гранитные блоки, находящиеся с краю (как слева, так и справа). Они хотят вытащить из-под лавочки как можно больше блоков так, чтобы она при этом не упала (передвигать оставшиеся блоки нельзя). Определите, какие блоки они должны оставить.

Формат входных данных

В первой строке входного файла записаны два числа: L – длина лавочки и K – количество гранитных блоков-ножек. Оба числа натуральные и не превышают 10 000.

Во второй строке записано K различных целых неотрицательных чисел, задающих положение каждой ножки. Положение ножки определяется расстоянием от левого края плиты до левого края ножки (ножка – это куб размером $1 \times 1 \times 1$). Ножки перечислены слева направо (то есть начиная с ножки с меньшим расстоянием до левого края)

Формат выходных данных

В выходном файле требуется перечислить ножки, которые грабителям нужно оставить. Для каждой ножки нужно выдать ее положение, как оно задано во входном файле. Ножки следует перечислять слева направо, как они встречаются во входном файле.

Примеры

bin.txt	bout.txt
5 2 0 2	2
13 4 1 4 8 11	4 8
14 6 1 6 8 11 12 13	6 8

Второй пример соответствует лавочке на рисунке.

Задача С. Выборы

Имя входного файла:	cin.txt
Имя выходного файла:	cout.txt
Максимальное время работы на одном тесте:	3 секунды
Максимальный объем используемой памяти:	16 мегабайт

На выборах в Государственную думу в избирательные бюллетени внесено N партий. Электронный сканер для считывания информации с бюллетеней передает информацию о каждом бюллетене в следующем формате: если в соответствующей клетке бюллетеня стоит пометка, то сканер передает + (плюс), в противном случае он передает – (минус). Таким образом, он передает последовательность из N символов – плюсов и минусов.

Бюллетень считается действительным, если пометка есть ровно в одной клетке. Недействительные бюллетени в подсчете результатов выборов не участвуют.

Партия проходит в Государственную Думу, только если она набирает не менее 7% от общего числа *действительных* бюллетеней.

Требуется вывести номера (в порядке их перечисления в бюллетене) всех партий, которые проходят в Государственную Думу.

Формат входных данных

В первой строке записано два числа, разделенные пробелом: N – количество партий и M – количество бюллетеней. Оба числа натуральные, $N \leq 200$, $M \leq 100\,000$.

В следующих M строках записана информация, полученная из бюллетеней. Каждая строка – последовательность из N символов + или – (без пробелов).

Гарантируется, что есть хотя бы один действительный бюллетень.

Формат выходных данных

В выходной файл выведите через пробел номера партий, прошедших в Думу, в порядке возрастания. Если ни одна из партий не проходит в Думу, то оставьте выходной файл пустым.

Система оценки

Программы, правильно работающие для случая $M \leq 100$ будут набирать не менее 70 баллов.

Примеры

cin.txt	cout.txt
3 4 +-- +-- -+- +++	1 2
1 5 + - - - -	1

Задача D. Реклама

Имя входного файла:	din.txt
Имя выходного файла:	dout.txt
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта

Фирма NNN решила транслировать свой рекламный ролик в супермаркете XXX. Однако денег, запланированных на рекламную кампанию, хватило лишь на две трансляции ролика в течение одного рабочего дня.

Фирма NNN собрала информацию о времени прихода и времени ухода каждого покупателя в некоторый день. Менеджер по рекламе предположил, что и на следующий день покупатели будут приходить и уходить ровно в те же моменты времени.

Помогите ему определить моменты времени, когда нужно включить трансляцию рекламных роликов, чтобы как можно большее количество покупателей прослушало ролик целиком от начала до конца хотя бы один раз. Ролик длится ровно 5 единиц времени. Трансляции роликов не должны пересекаться, то есть начало второй трансляции должно быть хотя бы на 5 единиц времени позже, чем начало первой.

Если трансляция ролика включается, например, в момент времени 10, то покупатели, пришедшие в супермаркет в момент времени 10 (или раньше) и уходящие из супермаркета в момент 15 (или позднее) успеют его прослушать целиком, а, например, покупатель, пришедший в момент времени 11, равно как и покупатель, уходящий в момент 14 — не успеют. Если покупатель успеет услышать только конец первой трансляции ролика (не сначала) и начало второй трансляции (не до конца), то считается, что он не услышал объявления. Если покупатель успеет услышать обе трансляции ролика, то при подсчете числа людей, прослушавших ролик, он все равно учитывается всего один раз (фирме важно именно количество различных людей, услышавших ролик).

Формат входных данных

Во входном файле записано сначала число N — количество покупателей ($1 \leq N \leq 300$). Далее записано N пар натуральных чисел — времена прихода и ухода людей. Все времена — натуральные числа, не превышающие 10^9 . Время ухода человека из супермаркета всегда строго больше времени его прихода в супермаркет.

Формат выходных данных

В выходной файл выведите три числа через пробел: количество людей, которые прослушают ролик целиком от начала до конца хотя бы один раз, и моменты времени, когда должна начинаться трансляция ролика. Моменты времени должны быть выведены в возрастающем порядке и должны быть натуральными числами, не превышающими $2 \cdot 10^9$. Если вариантов ответа несколько, выведите любой из них.

Система оценки

Программы, правильно работающие для случая, когда времена прихода и ухода всех покупателей не превышают 100, будут набирать не менее 60 баллов.

Примеры

din.txt	dout.txt	Пояснение
4 1 11 1 3 6 15 1 6	3 1 6	Трансляция роликов начинается в моменты времени 1 и 6. Первое объявление успевают прослушать покупатели номер 1 и 3, второе — 1 и 4. Когда бы ни начиналась трансляция объявления, 2-й покупатель не сможет его прослушать, так как находится в супермаркете менее 5 минут. Приведенный ответ является не единственным верным ответом на этот тест.
1 1 10	1 3 25	Объявление, трансляция которого начинается в момент 3, единственный покупатель обязательно услышит. Вторую трансляцию (раз она оплачена) мы можем сделать когда угодно, например, в 25 минут в пустом супермаркете (впрочем, мы не можем начать трансляцию второго объявления, например, в момент 7 — т.к. к этому моменту еще не закончится первая трансляция)
3 1 10 11 20 21 30	2 1 22	Объявление услышат лишь 2 из 3-х покупателей.

Задача Е. Ставки

Имя входного файла: `ein.txt`
 Имя выходного файла: `eout.txt`
 Максимальное время работы на одном тесте: 1 секунда
 Максимальный объем используемой памяти: 64 мегабайта

Перед началом тараканьих бегов всем болельщикам было предложено сделать по две ставки на результаты бегов. Каждая ставка имеет вид "Таракан №А придет раньше, чем таракан №В".

Организаторы бегов решили выяснить, могут ли тараканы прийти в таком порядке, чтобы у каждого болельщика сыграла ровно одна ставка из двух (то есть чтобы ровно одно из двух утверждений каждого болельщика оказалось верным). Считается, что никакие два таракана не могут прийти к финишу одновременно.

Формат входных данных

В первой строке входного файла записаны два натуральных числа через пробел: число K , не превосходящее 10, — количество тараканов и число N , не превосходящее 100, — количество болельщиков. Все тараканы пронумерованы числами от 1 до K . Каждая из следующих N строк содержит 4 натуральных числа A, B, C, D , не превосходящих K , разделенных пробелами. Они соответствуют ставкам болельщика "Таракан №А придет раньше, чем таракан №В" и "Таракан №С придет раньше, чем таракан №D".

Формат выходных данных

Если завершить бега так, чтобы у каждого из болельщиков сыграла ровно одна из двух ставок, можно, то следует вывести номера тараканов в том порядке, в котором они окажутся в итоговой таблице результатов (сначала номер таракана, пришедшего первым, затем номер таракана, пришедшего вторым и т. д.) в одну строку через пробел. Если таких вариантов несколько, выведите любой из них.

Если требуемого результата добиться нельзя, выведите одно число 0.

Примеры

<code>ein.txt</code>	<code>eout.txt</code>
3 2 2 1 2 3 1 2 3 2	3 2 1
3 4 1 2 1 3 1 2 3 1 1 2 2 3 1 2 3 2	0