

Задача А. Автобусы

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Для заезда в детский 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 входного файла

На вход программы поступают 3 натуральных числа, записанных через пробел - N , M и K , каждое из них не превосходит 10 000.

Формат выходного файла

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

Пример входного и выходного файлов

Входные данные	Выходные данные
10 4 7	2
10 4 5	0

Задача В. Лавочки

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 1 секунда
Ограничение по 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 входного файла

В первой строке входных данных содержатся два числа: L - длина лавочки и K - количество гранитных блоков-ножек. Оба числа натуральные и не превышают 10 000.

Во второй строке следуют 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 выходного файла

Требуется перечислить ножки, которые грабителям нужно оставить. Для каждой ножки нужно выдать ее положение, как оно задано во входных данных. Ножки следует перечислять слева направо, в том порядке, в котором они встречаются во входных данных.

Примеры входных и выходных файлов

Входные данные	Выходные данные
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: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

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 входного файла

В первой строке входных данных содержатся 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 выходного файла

Выведите через пробел номера партий, прошедших в Думу, в порядке возрастания. Если ни одна из партий не проходит в Думу, выводить ничего не нужно.

Примеры входных и выходных файлов

Входные данные	Выходные данные
3 4 +-- +-- -+- +--+	1 2
1 5 + - - - -	1

Задача D. Билет на электричку

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

В новых элитных электричках каждому пассажиру положено сидячее место. Естественно, количество сидячих мест ограничено и на всех их может не хватить. Маршрут электрички проходит через N станций, пронумерованных от 0 до $N-1$. Когда человек хочет купить билет, он называет два числа x и y – номера станций, откуда и куда он хочет ехать. При наличии хотя бы одного сидячего места на этом участке на момент покупки ему продается билет, иначе выдается сообщение «билетов нет» и билет не продается. Ваша задача – написать программу, обслуживающую такого рода запросы в порядке их прихода.

Формат входного файла

В первой строке содержатся три числа N – количество станций ($1 \leq N \leq 10\,000$), K – количество мест в электричке ($1 \leq K \leq 1000$) и M – количество запросов ($1 \leq M \leq 50\,000$). В следующих M строках описаны запросы, каждый из которых состоит из двух чисел x и y ($0 \leq x < y < N$).

Формат выходного файла

На каждый запрос ваша программа должна выдавать результат в виде числа 0 если билет не продается и 1 если билет был продан. Каждый результат должен быть на отдельной строке

Примеры входных и выходных файлов

Входные данные	Выходные данные
5 2 4	1
0 4	1
1 2	0
1 4	1
2 4	

Задача Е. Камни

Имя входного файла: input.txt
Имя выходного файла: output.txt
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На столе лежат N камней. За ход игрок может взять

- 1 или 2 камня, если N делится на 3;
- 1 или 3, если N при делении на 3 дает остаток один;
- 1, 2 или 3, если N при делении на 3 дает остаток два.

Каждый ход можно сделать при наличии достаточного количества камней. Проигрывает тот, кто хода сделать не может.

Формат входного файла

Вводится целое число $0 < N \leq 100$.

Формат выходного файла

Выведите 1 или 2 – номер игрока, который выиграет при правильной игре.

Примеры входных и выходных файлов

Входные данные	Выходные данные
3	2
5	1