

В соревновании **десять** задач. Больше задачи добавляться не будут.
Задачи можно сдавать до 23:59 10 января. По каждой задаче засчитывается
посылка, набравшая больше всего баллов.

Задача А. Разные кучки

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

На столе лежат конфеты. Изначально они разложены по четырём кучкам, в каждой кучке — целое положительное число конфет. За одну операцию можно взять любую кучку на столе и разделить её на две непустые кучки — в каждой должно также оказаться целое положительное число конфет. Можно ли, совершив ноль или более операций, добиться того, чтобы во всех кучках было разное число конфет?

Формат входных данных

В первой строке записаны через пробел четыре целых числа — количества конфет в изначальных четырёх кучках. Гарантируется, что эти числа лежат в пределах от 1 до 10 включительно.

Формат выходных данных

Выведите «YES», если можно сделать так, чтобы во всех кучках оказалось разное число конфет, или «NO», если это не удастся.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1 2 3 5	YES
2 4 4 2	NO
10 5 5 5	YES

Система оценки

В этой задаче три примера и 20 основных тестов. Каждый тест проверяется отдельно и даёт 5 баллов за правильный ответ.

Задача В. Как уравнивать числа

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Два целых положительных числа, a и b , загаданы и держатся в секрете. Известно лишь, что $1 \leq a, b \leq 10^9$ и $a \neq b$.

С числами можно делать в любом порядке следующие операции:

- выбрать целое положительное число x и умножить число a на x ;
- выбрать целое положительное число x и умножить число b на x .

После каждой операции сообщается результат сравнения a и b .

Сделайте числа a и b равными не более чем за 5000 операций.

Протокол взаимодействия

Это интерактивная задача. В каждом тесте начальные значения чисел a и b зафиксированы заранее. Гарантируется, что $1 \leq a, b \leq 10^9$ и $a \neq b$.

Решение выводит операции, каждую в отдельной строке. Операция записывается буквой, за которой следуют пробел и число:

- «a x »: умножить a на x ;
- «b x »: умножить b на x .

Число x должно быть целым, $1 \leq x \leq 2 \cdot 10^{18}$.

В ответ на каждую операцию решение получает строку с одним символом:

- «<», если $a < b$;
- «>», если $a > b$;
- «=», если $a = b$.

В последнем случае решение должно завершить работу. В остальных случаях можно выводить следующую операцию.

Если за 5000 операций не удалось сделать числа равными, решение должно завершить работу — тогда оно получит вердикт «Wrong Answer», а не какой-либо ещё.

Помните, что после вывода каждой операции нужно очищать буфер вывода (например, функцией «`fflush(stdout)`» в языке C, «`cout.flush()`» в C++ или «`sys.stdout.flush()`» в Python). В противном случае выведенная операция может не дойти до проверяющей программы, и тогда решение получит вердикт «Idleness Limit Exceeded».

Пример

Пустые строки добавлены для удобства чтения.

стандартный ввод	стандартный вывод	секретные значения
		$a = 3, \quad b = 5$
>	a 2	$a = 6, \quad b = 5$
<	b 3	$a = 6, \quad b = 15$
>	a 5	$a = 30, \quad b = 15$
=	b 2	$a = 30, \quad b = 30$

Система оценки

В этой задаче три подзадачи. Подзадача считается решённой, когда пройдены все тесты, подходящие под её ограничения. Подзадачи отличаются ограничениями на начальные значения a и b .

В первой подзадаче $1 \leq a, b \leq 10$, и за неё можно получить 30 баллов.

Во второй подзадаче $1 \leq a, b \leq 10^4$, и за неё можно получить ещё 40 баллов.

В третьей подзадаче $1 \leq a, b \leq 10^9$, и за неё можно получить оставшиеся 30 баллов.

Задача С. Два пропавших числа

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

У Глаши была последовательность из $n \geq 4$ элементов: каждое целое число от 1 до n встречалось в ней ровно один раз. Глаша взяла эту последовательность и стёрла из неё два последних элемента. Потом она разделила оставшуюся последовательность на две непустые части — левую и правую. В понедельник Глаша написала на доске левую часть последовательности, а во вторник — правую часть.

Гриша хочет узнать, какие числа Глаша стёрла. В понедельник он смотрит на левую часть последовательности, но может запомнить ко вторнику не очень большое количество информации. Во вторник Гриша видит правую часть последовательности, а также запомненную информацию с понедельника.

Напишите программу, которая поможет Грише узнать стёртые числа.

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. В каждом тесте все действия Глаши зафиксированы заранее. В конце каждой строки входных данных следует символ перевода строки.

Первый запуск

При первом запуске решение получает левую часть последовательности. В первой строке записано слово «first». Во второй строке записаны целые числа n и k_1 — исходная длина последовательности и длина левой части ($4 \leq n \leq 300\,000$, $0 < k_1 < n - 2$). В третьей строке записаны через пробел k_1 чисел — левая часть последовательности (все числа целые от 1 до n , все они различны).

Выведите в отдельной строке памятку — информацию, которую Гриша должен запомнить до вторника. Эта памятка должна иметь длину от 0 до 1000 символов и состоять из символов с ASCII-кодами от 32 до 126. Никаких других ограничений на содержимое памятки нет.

Второй запуск

При первом запуске решение получает памятку с понедельника, а также правую часть последовательности. В первой строке записано слово «second». Во второй строке дана памятка — ровно то, что решение выве-

ло при первом запуске. В третьей строке записаны целые числа n и k_2 — исходная длина последовательности и длина правой части (n такое же, как при первом запуске, $0 < k_2 < n$ и $k_1 + k_2 = n - 2$). В четвёртой строке записаны через пробел k_2 чисел — правая часть последовательности (все числа целые от 1 до n , все они различны и отличаются от чисел в левой части последовательности).

Выведите в отдельной строке два числа — те целые числа от 1 до n , которых нет ни в левой, ни в правой части последовательности. Числа можно выводить в любом порядке.

Пример

На каждом тесте входные данные при втором запуске зависят от того, что вывело решение при первом запуске.

Далее показаны два запуска какого-то решения на первом тесте.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
first 9 3 4 6 1	3 4 6 1

<i>стандартный ввод</i>	<i>стандартный вывод</i>
second 3 4 6 1 9 4 5 2 9 8	7 3

Система оценки

В этой задаче три подзадачи. Подзадача считается решённой, когда пройдены все тесты, подходящие под её ограничения.

В первой подзадаче $4 \leq n \leq 300$, и за неё можно получить 30 баллов.

Во второй подзадаче $4 \leq n \leq 3000$, и за неё можно получить ещё 30 баллов.

В третьей подзадаче $4 \leq n \leq 300\,000$, и за неё можно получить оставшиеся 40 баллов.

Задача D. Два квадрата

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Задано целое положительное число s . Выберем два целых положительных числа d_1 и d_2 . Рассмотрим два квадрата со сторонами d_1 и d_2 . Вычислим сумму их площадей t .

Выберите d_1 и d_2 так, чтобы площадь t отличалась от числа s как можно меньше, то есть величина $|s - t|$ была минимально возможной.

Формат входных данных

В первой строке записано целое число s ($1 \leq s \leq 10^{12}$).

Формат выходных данных

Выведите два целых положительных числа d_1 и d_2 так, чтобы минимизировать величину $|s - t|$. Если возможных ответов несколько, выведите любой из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4	1 2
15	3 2

Система оценки

В этой задаче две подзадачи. Подзадача считается решённой, когда пройдены все тесты, подходящие под её ограничения.

В первой подзадаче $1 \leq s \leq 10^6$, и за неё можно получить 40 баллов.

Во второй подзадаче $1 \leq s \leq 10^{12}$, и за неё можно получить оставшиеся 60 баллов.

Задача Е. Различные слагаемые

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В первой подзадаче $1 \leq n \leq 100$, и за неё можно получить 20 баллов.
Во второй подзадаче $1 \leq n \leq 5000$, и за неё можно получить ещё 50 баллов.
В третьей подзадаче $1 \leq n \leq 500\,000$, и за неё можно получить оставшиеся 30 баллов.

Задано целое положительное число n . Сколько существует способов представить его в виде суммы различных целых положительных слагаемых? Способы, отличающиеся только порядком слагаемых, считаются одинаковыми.

Поскольку ответ может быть очень большим, найдите остаток от деления искомого количества на простое число 998 244 353.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 500\,000$).

Формат выходных данных

Выведите одно целое число: количество способов представить n в виде суммы различных целых положительных слагаемых, взятое по модулю 998 244 353.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>пояснение</i>
5	3	$5 = 1 + 4$ $5 = 2 + 3$ $5 = 5$
10	10	$10 = 1 + 2 + 3 + 4$ $10 = 1 + 2 + 7$ $10 = 1 + 3 + 6$ $10 = 1 + 4 + 5$ $10 = 1 + 9$ $10 = 2 + 3 + 5$ $10 = 2 + 8$ $10 = 3 + 7$ $10 = 4 + 6$ $10 = 10$

Система оценки

В этой задаче три подзадачи. Подзадача считается решённой, когда пройдены все тесты, подходящие под её ограничения.

Задача F. Проекция и тень

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Представим себе трёхмерное пространство, в котором зафиксирован бесконечно далёкий точечный источник света: световые лучи идут параллельно. Светлана поместила в это пространство невырожденный треугольник ABC . Затем она отметила на плоскости Oxy проекцию этого треугольника, а на плоскости Oxz — его тень. Получились два невырожденных треугольника. После этого Светлана убрала треугольник ABC и выключила свет.

Даны координаты вершин проекции и координаты вершин тени. Восстановите треугольник ABC .

Формат входных данных

В первых трёх строках задана проекция треугольника на плоскость Oxy . В следующих трёх строках задана тень треугольника на плоскости Oxz .

Каждое из этих описаний занимает по три строки. В первых трёх строках заданы по порядку проекции вершин A, B, C на плоскость Oxy . Во вторых трёх строках заданы по порядку тени вершин A, B, C на плоскости Oxz . Каждая вершина описывается тремя целыми числами x, y, z — координатами по соответствующим осям ($0 \leq x, y, z \leq 100$).

Координаты z вершин проекции равны нулю. Координаты y вершин тени также равны нулю. Все остальные координаты — **строго положительные**. Кроме того, проекция и тень — невырожденные треугольники.

Гарантируется, что существуют треугольник со **строго положительными** вещественными координатами и направление световых лучей, при которых получаются ровно такие проекция и тень.

Формат выходных данных

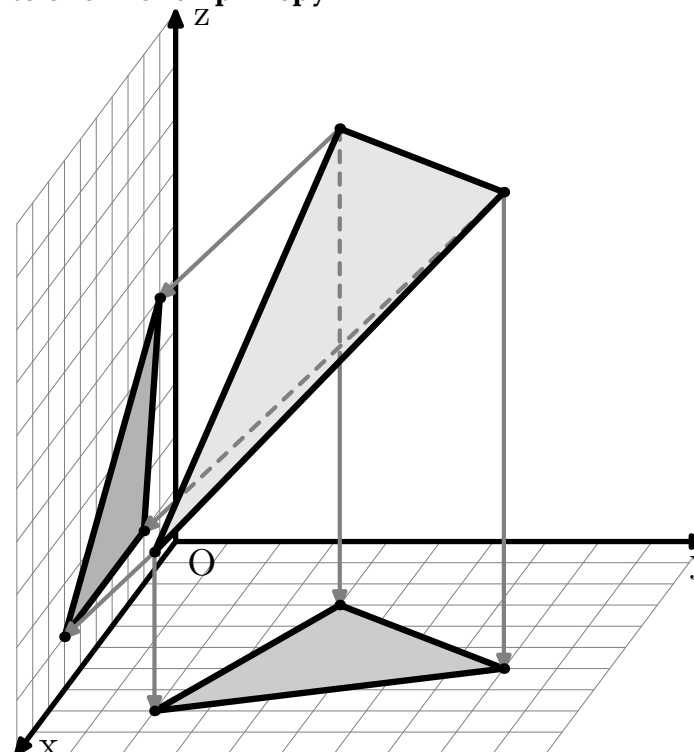
Выведите три строки по порядку: координаты вершин A, B, C исходного треугольника. Каждую вершину опишите тремя **строго положительными** вещественными числами x, y, z — координатами по соответствующим осям ($10^{-5} \leq x, y, z \leq 10^5$).

Если возможных треугольников ABC несколько, выведите любой из них. Ответ считается верным, если существует направление световых лучей, при котором вершины проекции и вершины тени находятся на расстоянии не более 10^{-2} от соответствующих вершин, заданных во входных данных.

Пример

стандартный ввод	стандартный вывод
8 2 0	8.0 2.0 3.0
6 8 0	6.0 8.0 9.0
3 4 0	3.0 4.0 9.0
7 0 1	
2 0 1	
1 0 5	

Пояснение к примеру



Система оценки

В этой задаче один пример и 25 основных тестов. Каждый тест проверяется отдельно и даёт 4 балла за правильный ответ.

Задача G. Составление слов

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Дан словарь английских слов. Кроме того, дано слово w из этого словаря. Какие слова из словаря можно составить, используя буквы слова w ?

Слово x можно составить из букв слова w , если каждая буква алфавита встречается в w не меньше раз, чем в x .

В каждом тесте может быть задано несколько слов w .

Формат входных данных

В первой строке задано целое число n — количество слов в словаре ($n = 173\,528$). В следующих n строках записаны слова в словаре, по одному на строке. Слова следуют в лексикографическом порядке.

В следующей строке задано целое число k — количество вопросов ($k \geq 1$). Каждая из следующих k строк содержит один вопрос — слово w , для которого нужно решить задачу. Гарантируется, что это слово содержится в словаре, а также что все k вопросов различны.

Все слова состоят из маленьких английских букв.

Во всех тестах словарь одинаковый. Это свободно распространяемый словарь ENABLE для игр со словами на английском языке.

Формат выходных данных

Для каждого вопроса w сначала в отдельной строке выведите целое число p — количество слов, которые можно составить из букв слова w . В следующих p строках выведите сами эти слова в лексикографическом порядке, по одному слову на строке.

Гарантируется, что всего в правильном ответе не более 100 000 слов, то есть сумма всех чисел p не превосходит 100 000.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
173528	4
aa	ay
aah	ya
aahed	yay
<...>	zyzzyva
zyzzyva	7
zyzzyvas	be
2	best
zyzzyva	bet
best	bets
	es
	et
	set

Пояснение к примеру

В примере выше словарь сокращён. Полную версию примера можно скачать отдельно здесь:

<https://tsweb.ru/olymp/2022-2023/sub-words.unix.txt>

с переводами строк для Unix или

<https://tsweb.ru/olymp/2022-2023/sub-words.windows.txt>

с переводами строк для Windows.

Система оценки

В этой задаче четыре подзадачи. Подзадача считается решённой, когда пройдены все тесты, подходящие под её ограничения.

В первой подзадаче общее количество слов в правильном ответе не превосходит 100, и за неё можно получить 25 баллов.

Во второй подзадаче общее количество слов в правильном ответе не превосходит 1000, и за неё можно получить 25 баллов.

В третьей подзадаче общее количество слов в правильном ответе не превосходит 10 000, и за неё можно получить 25 баллов.

В четвёртой подзадаче общее количество слов в правильном ответе не превосходит 100 000, и за неё можно получить 25 баллов.

Задача Н. Найдите две точки

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

На плоскости загаданы две точки P и Q . Их положение держится в секрете: известно лишь, что они различны и имеют целые координаты от 0 до 10^6 включительно.

Чтобы найти секретные точки, можно задавать вопросы. Каждый вопрос — это какая-то целая точка R на плоскости. В ответ мы узнаём манхэттенское расстояние от R до ближайшей из точек P и Q . Говоря формально, мы получаем в ответ число $\min\{d(R, P), d(R, Q)\}$, где расстояние вычисляется как $d(A, B) = |x_A - x_B| + |y_A - y_B|$.

Найдите обе секретные точки.

Протокол взаимодействия

Это интерактивная задача.

В каждом тесте положение двух секретных точек зафиксировано заранее. Координаты секретных точек — целые числа от 0 до 10^6 включительно. Гарантируется, что секретные точки различны.

В каждый момент у решения есть выбор: либо задать вопрос, либо завершить работу.

Чтобы задать вопрос, выведите в отдельной строке координаты точки — два целых числа от -10^9 до $+10^9$ включительно. В ответ решение получит строку с одним целым числом — манхэттенским расстоянием от точки в вопросе до ближайшей из секретных точек. Чтобы найти секретную точку, нужно назвать её в вопросе — и получить в ответ расстояние 0.

Когда решение корректно завершает работу вместо того, чтобы задать очередной вопрос, проверяется, что обе секретные точки найдены хотя бы по разу. Если это не так, решение получает вердикт «Wrong Answer».

Прямого ограничения на количество вопросов нет. Разрешается повторять вопросы. Разрешается задавать вопросы после того, как обе секретные точки найдены.

Помните, что после вывода каждого вопроса нужно выводить **перевод строки**, а затем **очищать буфер** вывода: например, функцией «`fflush(stdout)`» в языке C, «`cout.flush()`» в C++ или «`sys.stdout.flush()`» в Python. Кроме того, следует **прочитать ответ** на предыдущий вопрос, прежде чем задавать следующий. В противном случае

очередной вопрос может не дойти до проверяющей программы, и тогда решение получит вердикт «Idleness Limit Exceeded».

Пример

Пустые строки добавлены для удобства чтения.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
	2 2
4	
	4 4
3	
	6 2
2	
	5 1
0	
	5 1
0	
	4 7
0	
	9 9
7	

Система оценки

В этой задаче две подзадачи. Подзадача считается решённой, когда пройдены все тесты, подходящие под её ограничения.

В первой подзадаче координаты секретных точек — целые числа от 0 до 10^4 включительно, и за неё можно получить 30 баллов.

Во второй подзадаче координаты секретных точек — целые числа от 0 до 10^6 включительно, и за неё можно получить оставшиеся 70 баллов.

Задача I. Цифры по очереди

Имя входного файла:	<i>стандартный ввод</i>
Имя выходного файла:	<i>стандартный вывод</i>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Квентин и Ксения показывают арифметический фокус с делимостью. Для фокуса используется доска с n листами бумаги, прикреплёнными к ней в ряд слева направо.

Сначала на сцене появляется Квентин. Кто-то из зрителей загадывает двузначное число d . Квентин выписывает n -значное число x , записывая слева направо по одной цифре на листе. Зрители убеждаются, что число x делится на d без остатка.

Потом Квентин переворачивает все листы — так, что цифры не видны, — и уходит, а на сцену выходит Ксения. Ксения не знает чисел d и x . Она проходит вдоль n листов слева направо. Нечётные листы — первый, третий, пятый, ... — она переворачивает, чтобы открыть цифру, написанную Квентином. А на чётных листах — втором, четвёртом, ... — сама пишет цифру, не глядя на обратную сторону листа. Когда Ксения заканчивает, публика убеждается, что n -значное число, получившееся на месте числа x , также делится на d без остатка!

Напишите программу, которая показывает этот фокус.

Протокол взаимодействия

В этой задаче ваше решение будет запущено на каждом тесте два раза. В конце каждой строки входных данных следует символ перевода строки.

Первый запуск

При первом запуске решение действует за Квентина. В первой строке записано имя «*Quentin*». Во второй строке записано целое число n — количество листов бумаги, прикреплённых к доске ($5 \leq n \leq 8$). В третьей строке записано целое двузначное число d — делитель ($10 \leq d \leq 99$).

Выведите одно целое n -значное число x , делящееся на d . Первая цифра не может быть нулём.

Второй запуск

При втором запуске решение действует за Ксению. В первой строке записано имя «*Xenia*». Во второй строке записано целое число n — количество листов бумаги, прикреплённых к доске, то же, что и при первом запуске.

Далее следует интерактивная часть взаимодействия. Ксения проходит вдоль листов с цифрами слева направо. Для каждой цифры на нечётной позиции — первой, третьей, пятой, ... — решение получает строку с цифрой числа x , написанной на этой позиции. Для каждой цифры на чётной позиции — второй, четвёртой, ... — выведите строку с цифрой (от 0 до 9), которую следует написать на этой позиции. Полученное в итоге n -значное число должно делиться на число d из первого запуска.

Помните, что после вывода каждой цифры нужно выводить перевод строки и очищать буфер вывода: например, функцией «`fflush(stdout)`» в языке C, «`cout.flush()`» в C++ или «`sys.stdout.flush()`» в Python. В противном случае очередная цифра может не дойти до проверяющей программы, и тогда решение получит вердикт «`Idleness Limit Exceeded`».

Пример

Далее показаны два запуска какого-то решения на первом тесте. Входные данные при втором запуске зависят от того, что вывело решение при первом запуске. Пустые строки добавлены для удобства чтения — при реальном запуске их не будет.

<i>стандартный ввод</i>	<i>стандартный вывод</i>
Quentin 8 25	94624850

<i>стандартный ввод</i>	<i>стандартный вывод</i>
Xenia 8 9 6 4 5	 7 2 9 0

Пояснение к примеру

Легко проверить, что число $x = 94\,624\,850$ делится на $d = 25$.

Легко проверить также, что полученное при втором запуске число $97\,624\,950$ тоже делится на 25.

Система оценки

В этой задаче четыре подзадачи. Подзадача считается решённой, когда пройдены все тесты, подходящие под её ограничения, а также пример.

В первой подзадаче $n = 8$, и за неё можно получить 40 баллов.

Во второй подзадаче $n = 7$, и за неё можно получить 30 баллов.

В третьей подзадаче $n = 6$, и за неё можно получить 20 баллов.

В четвёртой подзадаче $n = 5$, и за неё можно получить 10 баллов.

Задача J. Доминошки и полюса

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Рассмотрим прямоугольное поле из $h \times w$ клеток. Поле замощено доминошками — фигурами из двух клеток с общей стороной: каждая клетка покрыта ровно одной доминошкой, и ни одна доминошка не выходит за пределы поля. Вот пример поля, на котором каждая доминошка обозначена своей буквой:

A	B	B	C
A	X	X	C
D	D	Y	Y

У каждой доминошки есть два полюса: клетка со знаком «+» и клетка со знаком «-». Для каждой доминошки можно выбрать ориентацию: какая из её клеток содержит плюс, а какая — минус. Вот пример итоговой расстановки плюсов и минусов для поля выше:

+	+	-	-
-	-	+	+
+	-	-	+

В итоговой расстановке *штраф* — это сумма двух величин: количества пар клеток с общей стороной, содержащих «+», и количества пар клеток с общей стороной, содержащих «-». В примере выше штраф равен $3 + 4 = 7$.

Задано замощение поля доминошками. Выберите ориентацию для доминошек так, чтобы штраф был как можно меньше.

Формат входных данных

В первой строке заданы два целых числа h и w : высота и ширина поля ($1 \leq h \leq 5$, $1 \leq w \leq 6$). Следующие h строк содержат по w больших английских букв и задают замощение поля доминошками.

Гарантируется, что каждая буква английского алфавита либо не встречается на поле, либо встречается ровно два раза в клетках, имеющих общую сторону.

Формат выходных данных

Выведите h строк, в каждой по w символов: расстановку плюсов и минусов, минимизирующую штраф. В каждой доминошке должен быть один плюс и один минус. Если существует несколько оптимальных расстановок, выведите любую из них.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 3 ABV ACC	-+- +-+

Система оценки

В этой задаче один пример и 20 основных тестов. Каждый тест проверяется отдельно и даёт 5 баллов за правильный ответ.