Общая информация по задачам олимпиады

Ограничение по памяти

Во всех задачах ограничение составляет 512 МБ.

Ограничение на размер исходного кода программы

Во всех задачах размер файла с исходным кодом решения не должен превышать 256 КБ.

Ограничение на посылку решений

По каждой задаче на проверку принимается не более 50 решений.

По каждой задаче участник не может отправить решение более одного раза в течение 30 секунд. Это ограничение не распространяется на последние 15 минут соревнований.

Система оценки

Каждая задача олимпиады поделена на несколько подзадач. Чтобы набрать баллы по подзадаче, программа должна пройти все тесты этой подзадачи.

За каждую задачу выставляется суммарный балл по всем ее подзадачам. В каждой подзадаче оценивается лучшее решение, то есть за подзадачу выставляется максимальный набранный по ней балл среди всех решений.

Получение информации о результатах проверки

Чтобы получить информацию о проверке вашего решения, используйте ссылку «Информация о проверке» во вкладке «Решения» в PCMS2 Web Client. По каждой задаче вам будет доступна информация по количеству набранных баллов в каждой подзадаче или результат проверки на первом непройденном тесте.

Таблица результатов

Во время соревнования доступна текущая таблица результатов. Для доступа к ней используйте ссылку «Результаты» в PCMS2 Web Client. Таблица результатов в PCMS2 Web Client не является окончательной.

Задача A. Sheet Metal

Ограничение по времени: 2 секунды

У Джеймса есть прямоугольный металлический лист шириной w_1 и длиной h_1 миллиметров. Такой огромный лист сложно и тяжело передвигать, поэтому Джеймс со своими друзьями желают сделать из листа куски поменьше.

Они нашли станок, который им в этом поможет. У станка есть форма, имеющую вид прямоугольника w_2 на h_2 миллиметров. Под эту форму можно положить лист. Форма не обязана быть везде над листом металла, разрешается, чтобы части формы были не над ним. Прямоугольный лист можно поворачивать, однако требуется, чтобы стороны обоих прямоугольников были параллельны осям координат.

После запуска станка, форма выжжет все, что находится под ней, и та часть листа, что лежала под формой, будет больше не пригодна. Все, что находится вне формы, Джеймсу и приятелям нужно забрать с собой. При этом, возможно, лист распадется на несколько несвязных частей. Если граница формы совпадает с границей листа, то будем считать, что в этом месте лист металла также будет несвязным.

Чтобы унести оставшися куски было проще всего, найдите такой способ положить лист на станок, чтобы **максимальная площадь среди всех оставшихся кусков была минимальной**. Если никаких кусков металла не остается, выведите 0.

Формат входных данных

Первая строка входного файла содержит целые числа w_1, h_1, w_2, h_2 ($1 \leqslant w_1, h_1, w_2, h_2 \leqslant 10^6$) — размеры металлического листа и формы.

Формат выходных данных

Выведите минимальную возможную площадь наибольшего из оставшихся кусков металла.

Система оценки

Подзадача	Баллы	Ограничения
1	50	$w_2 \leqslant h_2 \leqslant w_1 \leqslant h_1$
2	50	Без дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
2 3 1 3	1.5
2 3 3 3	0
2 3 2 2	1
2 3 3 1	1.5

Задача B. Lunchtime Fruits

Ограничение по времени: 2 секунды

В школьной столовой есть четыре типа фруктов: абрикосы, бананы, яблоки и груши. В меню на сегодня три разных вида полдника:

1. два абрикоса, один банан и одно яблоко;
2. два абрикоса и два яблока;
3. один абрикос, один банан, два яблока и одна груша.

Сотрудники хотят из имеющихся продуктов составить как можно больше полдников для детей. Помогите им это сделать!

Поскольку в ближайшие дни на склад будет приходить разное число фруктов каждого типа, вам необходимо решить задачу для нескольких сценариев количеств фруктов.

Формат входных данных

Первая строка входных данных содержит целое число t ($1 \leqslant t \leqslant 10^4$) — количество сценариев, для которых нужно решить задачу.

Следующие t строк содержат описание тестовых сценариев каждая. Каждая строка содержит четыре числа a, b, c и d ($1 \leqslant a, b, c, d \leqslant 10^9$) — количество абрикосов, бананов, яблок и груш соответственно.

Формат выходных данных

Для каждого тестового сценария выведите максимальное количество полдников, которое можно составить.

Система оценки

Подзадача	Баллы	Ограничения
1	10	$t \leqslant 10$; $a, b, c, d \leqslant 10$
2	20	$t \leqslant 10^4$; $a, b, c, d \leqslant 10$
3	20	$t \leqslant 10$; $a, b, c, d \leqslant 200$
4	20	$t \leqslant 10$; $a, b, c, d \leqslant 10^6$
5	30	Без дополнительных ограничений

Пример

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

Замечание

В первом сценарии можно сделать два полдника: один типа 1 и один типа 3.

Во втором сценарии подойдет набор полдников $(2, 3)$.

В третьем сценарии оптимальный набор $(1, 1)$.

В четвертом сценарии можно сделать три полдника типа 3.

Задача C. Sliding Dominoes

Ограничение по времени: 2 секунды

Сыграем в игру! Клеточное поле для игры состоит из n строк и m столбцов, **где n и m нечетны**. На поле лежат кости домино, каждая кость домино покрывает две соседние клетки по горазонтали или вертикали. В начале игры, каждая клетка, кроме одной, покрыта ровно одной доминошкой, а одна клетка пуста.

За один шаг можно *подвинуть* любую доминошку в направлении, параллельном ее положению, если клетка в этом направлении пуста. Двигать доминошки можно сколько угодно раз, и в любой момент можно остановиться.

У каждой клетки поля есть определенная стоимость (положительная или отрицательная). Когда мы передвигаем домино, клетка, которая была раньше покрыта домино, теперь становится свободной. Если эта клетка стала свободной впервые за ход игры, ее стоимость прибавляется к текущему счету.

Передвиньте домино таким образом, чтобы максимизировать конечный счет, то есть, суммарную стоимость клеток, открытых хотя бы раз в течение игры.

Формат входных данных

В первой строке содержатся два числа n и m — число строк и столбцов поля ($1 \leqslant n, m \leqslant 499$; n и m нечетны). Следующие n строк содержат по m символов каждая и описывают поле. Пустая клетка на поле обозначается точкой `.`, горизонтальное домино — парой символов `<` (левая клетка) и `>` (правая клетка), вертикальное домино — парой символов `^` (верхняя клетка) и `v` (нижняя клетка). Символы задают корректное замощение домино, на поле ровно одна пустая клетка.

Следующие n строк содержат по m чисел и описывают стоимости клеток. Стоимость каждой клетки — целое число от -1000 до 1000 включительно. Гарантируется, что пустая клетка на входном поле имеет стоимость 0.

Формат выходных данных

Выведите одно число — максимальную суммарную стоимость, которую можно собрать с открытых клеток.

Система оценки

Подзадача	Баллы	Ограничения
1	10	$n = 1$, $m \leqslant 3$
2	24	$n = 1$, $m \leqslant 499$
3	14	$n, m \leqslant 7$
4	23	$n, m \leqslant 499$, все стоимости неотрицательны
5	29	Без дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
1 5 <>.<> 5 2 0 9 -2	5
3 3 <>^ ^.v v<> 1 1 1 1 0 1 1 1 1	0

Задача D. Lost in Translation

Ограничение по времени: 2 секунды

Это задача с двойным запуском. Ваше решение будет запущено два раза.

Вам необходимо написать программу, которая передает данные по ненадежному каналу связи. На одном конце провода (во время первого запуска) вы получаете двоичную строку длины n, и должны уметь восстановить ее на другом конце провода (во время второго запуска).

К счастью, канал связи позволяет посылать строку с k различными типами символов ($k > 2$) и использовать строки длины m ($m > n$). Однако, в результате передачи строки по этому каналу, **все вхождения какого-то из k типов символов будут удалены**. Оставшиеся символы строки будут идти в том же порядке, как и раньше. Ваша задача состоит в том, чтобы придумать схему кодирования, позволяющую восстановить исходную строку во время второго запуска.

Для ускорения тестирования в одном тесте вам предстоит закодировать и передать сразу t строк. Удаление символов в этих строках будет независимым, в разных строчках могут быть удалены разные символы.

Формат входных данных

При первом запуске на первой строке ввода находится число 1. Следующая строка содержит целые числа t, n, m и k — число строк, которые необходимо закодировать, длина каждой из них, разрешенная длина строки, которую можно вывести, и число различных символов, которые можно использовать ($1 \leqslant t \leqslant 100$, $k = 3$ или $k = 4$).

Каждая из следующих t строк содержит строку длины n из нулей и единиц.

Если $k = 4$, то вы можете использовать для кодирования символы `A`, `B`, `C`, `D`. Если $k = 3$, то вы можете использовать только `A`, `B` и `C`.

При втором запуске на первой строке ввода находится число 2. Вторая строка также содержит целые числа t, n, m и k, такие же, как и в первом запуске. Далее следуют t строк, которые вывела ваша программа в первом запуске, но в каждой строке были удалены все символы какого-то одного типа. Строки, подающиеся на вход вашей программе во втором запуске, будут идти в том же порядке, как и в первом запуске.

Формат выходных данных

В первом запуске вам необходимо вывести t непустых строк. Каждая из должна состоять из не более, чем m символов из алфавита { `A`, `B`, `C`, `D` } или { `A`, `B`, `C` }, в зависимости от текущего k.

Символ для удаления будет выбран так, чтобы строка не стала пустой, например, в строке `CCCC` для удаления не будет выбран символ `C`.

При втором запуске раскодируйте все t строк и выведите исходные двоичные строки длины n.

Система оценки

Подзадача	Баллы	Ограничения
1	27	$t = 1$, $n = 10$, $k = 4$, $m = 20$
2	14	$t = 100$, $n = 100$, $k = 4$, $m = 200$
3	28	$t = 100$, $n = 100$, $k = 3$, $m = 200$
4	20	$t = 100$, $n = 100$, $k = 3$, $m = 190$
5	11	$t = 100$, $n = 100$, $k = 3$, $m = 180$

Пример

стандартный ввод	стандартный вывод
1 2 10 20 4 0111011001 1111111110	BAACBBACDCDDAACCAABD DABBADCBCBBCCACA
2 2 10 20 4 AACACDCDDAACCAAD DBBDCBCBBCCC	0111011001 1111111110

Замечание

В примере необходимо передать две строки: 0111011001 и 1111111110, используя строки длины $m = 20$ и $k = 4$ типа символов. Предположим, что в первом запуске программа вывела BAACBBACDCDDAACCAABD для первой строки и DABBADCBCBBCCACA для второй строки.

Перед вторым запуском программа жюри удалит один тип символов из каждой строки. Для примера, в первой строчке удалили все буквы B, а во второй — все буквы A. По этим данным необходимо восстановить изначальные двоичные строки 0111011001 и 1111111110.

Задача E. Summer School

Ограничение по времени: 2 секунды

Начался набор на новую смену Школы олимпиадного программирования. В этом году произошло изменение в образовательных параллелях. В школе будет n параллелей для уровней от 0 до $n-1$. В каждой параллели есть k мест для поступающих школьников.

Также по результатам участия в олимпиадах и тренировок каждый школьник получил оценку от искусственного интеллекта — целое число от 0 до $n-1$.

Школьнику с уровнем L будет полезным обучаться в параллели x, если уровень школьника отличается от уровня параллели не более чем на $d + L \cdot \frac{p}{100}$, то есть $|x - L| \leqslant d + L \cdot \frac{p}{100}$.

Каждый день в систему регистрации приходят заявки: школьники либо регистрируются, либо отказываются от участия. Чтобы помочь спланировать школьникам свое лето, организаторы решили дописать в систему программу, которая определяет, сколько школьникам участие будет полезным.

По случайному стечению обстоятельств каждый день происходит одно из двух:

1. `+ L v` — v школьников уровня L зарегистрировались;
2. `- L v` — v школьников уровня L отказались от участия.

Вам требуется написать программу, которая после каждого дня определит, какое максимальное число школьников можно зачислить, чтобы им смена оказалась полезной.

Формат входных данных

В первой строке заданы целые числа n, k, d и p — число параллелей, число мест в каждой параллели, параметры для определения полезности параллели, соответственно ($1 \leqslant n \leqslant 5 \cdot 10^5$, $1 \leqslant k \leqslant 10^9$, $0 \leqslant d \leqslant n$, $0 \leqslant p \leqslant 100$).

Во второй строке задано целое число m — число дней для обработки ($1 \leqslant m \leqslant 5 \cdot 10^5$).

В следующих m строках заданы события, которые происходят каждый день. В каждой строке задано либо `+ L v`, либо `- L v`, где L — уровень школьника, а v — число таких заявок ($0 \leqslant L < n$, $1 \leqslant v \leqslant 10^9$).

Формат выходных данных

В m строках выведите по целому числу, сколько школьников можно зачислить, чтобы всем зачисленным смена оказалась полезной.

Система оценки

Назовем C — максимальное число заявок одного уровня, которые были в какой-то момент зарегистрированы в системе.

Подзадача	Баллы	Ограничения
1	10	$n \leqslant 5$, $m \leqslant 10$, $C \leqslant 4$
2	10	$n \leqslant 30$, $m \leqslant 100$, $C \leqslant 30$
3	10	$n, m \leqslant 100$, $C \leqslant 10^6$
4	10	$n, m \leqslant 10^5$, $d = 0$
5	10	$n, m \leqslant 10^5$, $d \leqslant 1$
6	10	$n, m \leqslant 10^5$, $p = 0$
7	10	$n, m \leqslant 10^5$, $v = 1$
8	10	$n, m \leqslant 10^5$, только запросы вида ‘+’
9	10	$n, m \leqslant 10^5$
10	10	Без дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
5 2 1 25 5 + 4 7 - 4 3 + 2 5 + 3 5 - 3 2	6 4 8 8 8
5 2 1 1 6 + 0 4 + 1 3 - 0 2 + 3 7 + 4 1 - 3 6	4 6 5 10 10 7

Замечание

В первом примере школьники уровня 4 могут поступить в параллель 2, 3 и 4, поэтому после первого дня 6 школьников из 7 могут с пользой для себя поступить в смену.

Во втором примере все школьники могут поступить только в параллели такого же уровня, либо отличающиеся по уровню на единицу.