



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. Alphabet Contest

Ограничение по времени: 1 секунда

Выпускники детского сада участвуют в открытой олимпиаде на знание английского алфавита. В одном из заданий им нужно было правильно перечислить все буквы английского языка в алфавитном порядке без повторов.

Дети начинали называть буквы, не дожидаясь, пока закончит предыдущий участник, то есть параллельно. В это время учитель записывал всё сказанное на доске в одну общую для всех строку. Алфавит еще достаточно сложный для детей, и иногда они совершают ошибки в виде пропуска буквы. Несколько подряд пропущенных букв считаются за несколько ошибок. Известно, что всего детьми в этом задании было совершено не более k ошибок суммарно для всех. Если участник в какой-то момент уставал и совсем переставал перечислять буквы, не дойдя до конца алфавита, учитель не считал это за ошибки.

По числу k и получившейся строке найдите минимальное число детей, которые могли участвовать в олимпиаде, либо определите, что такой результат невозможен.

Формат входных данных

В первой строке задано одно целое число k ($0 \leq k \leq 1000$) — максимальное количество пропущенных букв. Во второй строке задана строка s из заглавных английских букв (длина строки s не превосходит 1000) — названные участниками буквы.

Формат выходных данных

Выведите в единственной строке одно слово «Impossible», если данные некорректны (то есть нельзя было получить такую строку, пропустив не более k букв), иначе выведите минимальное число участников олимпиады.

Система оценки

Подзадача	Баллы	Дополнительные ограничения
1	16	$k = 0$
2	16	в s буквы расположены в алфавитном порядке
3	11	s_i равно либо 'a', либо 'b'
4	11	строку можно получить, пропустив не более k букв, и минимальное число участников не более 3
5	46	нет дополнительных ограничений

Примеры

стандартный ввод	стандартный вывод
5 ABDVCBADB	4
100 INNOPOLIS	4
0 ABB	Impossible

Пояснения к примерам

В первом примере строки отдельно для каждого участника могли выглядеть следующим образом:

Участник 1: ABD — 1 ошибка;

Участник 2: BC — 1 ошибка;

Участник 3: VD — 2 ошибки;



Участник 4: АВ — 0 ошибок;

Во втором примере:

Участник 1: INO — 12 ошибок;

Участник 2: NOP — 13 ошибок;

Участник 3: LS — 17 ошибок;

Участник 4: I — 8 ошибок;



Задача В. Need More T-shirts!

Ограничение по времени: 1 секунда

Приближается Innopolis Open 2020. А это значит, что пора заказывать футболки для участников олимпиады.

Организаторы решили, что было бы здорово купить футболки разных цветов, чтобы можно было просто понять, кто является организатором, жюри или участником олимпиады.

Для этой работы было выделено два человека, которые должны были решить, какое количество футболок каждого цвета надо заказать. Однако они не договорились между собой, и один из них записывал число футболок определенного цвета, которое необходимо заказать, а другой — процент футболок определенного цвета. В итоге у них получился список длины n , где каждый элемент списка a_i — это либо число футболок цвета i , либо процент числа футболок цвета i от общего числа футболок. Причем известно, что в списке хотя бы один элемент a_i является числом футболок цвета i .

Теперь организаторам надо понять — сколько же всего футболок им нужно купить для олимпиады. Помогите им найти все возможные варианты общего числа футболок, для которого мог бы получиться такой список.

Формат входных данных

В первой строке записано одно целое число n — количество различных цветов ($1 \leq n \leq 10^5$).

Во второй строке записано n целых чисел a_i — либо число футболок, либо процент от общего числа футболок для цвета i ($1 \leq a_i \leq 10^9$).

Формат выходных данных

В первой строке выведите число вариантов для общего числа футболок, для которого мог бы получиться такой список.

Во второй строке все возможные варианты для общего числа футболок в отсортированном порядке.

Система оценки

Подзадача	Баллы	Ограничения
1	13	$1 \leq n \leq 15, 1 \leq a_i \leq 100$
2	21	$1 \leq n \leq 100, 1 \leq a_i \leq 100$
3	38	$1 \leq n \leq 100, 1 \leq a_i \leq 10^9$
4	28	$1 \leq n \leq 10^5, 1 \leq a_i \leq 10^9$

Примеры

стандартный ввод	стандартный вывод
2 1 50	2 2 51
3 20 30 70	4 120 125 140 300
4 2 40 90 5	3 137 470 840
7 10 20 5 15 30 17 3	1 100

Пояснение к примеру

Пояснение к первому примеру:



Если общее число футболок равно 2: одна цвета 1 и одна цвета 2, то мы можем получить такой список, если первый элемент (1) — это число футболок цвета 1, а второй элемент (50) — это процент футболок цвета 2: $2 = 1 + 2 \cdot \frac{50}{100}$.

Если общее число футболок равно 51: одна цвета 1 и 50 цвета 2, то мы можем получить такой список, если оба элемента — это число футболок цвета 1 и 2, соответственно: $51 = 1 + 50$.

Пояснение ко второму примеру:

$$120 = 20 + 30 + 70$$

$$125 = 125 \cdot \frac{20}{100} + 30 + 70$$

$$140 = 140 \cdot \frac{20}{100} + 140 \cdot \frac{30}{100} + 70$$

$$300 = 300 \cdot \frac{20}{100} + 30 + 300 \cdot \frac{70}{100}$$

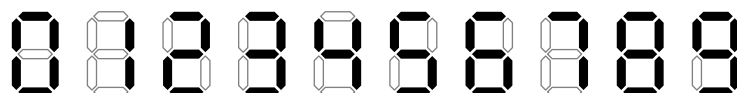


Задача С. The Final Countdown

Ограничение по времени: 2 секунды

Поговаривают, что в Иннополисе будут проводить IOI 2048! В связи с этим в Иннополисе установили большой дисплей, обозначающий количество наносекунд до начала олимпиады! Изначально на дисплее было установлено некоторое число, и в каждую наносекунду это число уменьшается на единицу. Ведущие нули не отображаются.

Цифры на этом дисплее показываются стандартными семисегментными индикаторами. То, как выглядят цифры, показано на картинке:



Наносекунды меняются настолько быстро, что увидеть число, которое сейчас показывает дисплей, практически невозможно. Но путем подключения к питанию дисплея высокоточного датчика удалось получить значения a_i — количество сегментов, включенных в каждую из n наносекунд подряд. Поскольку до IOI 2048 еще есть время, число на таймере в любой момент было положительное.

Напишите программу, которая вычисляет количество возможных начальных значений (соответствующих измерению a_1), а также любые m из этих значений. В случае если подходящих значений меньше m , следует вывести их все.

Формат входных данных

Первая строка содержит два числа n и m ($1 \leq n \leq 10^5$, $0 \leq m \leq 10$) — количество наносекунд и интересующее количество вариантов. Следующая строка содержит n целых чисел a_i ($2 \leq a_i \leq 1000$) — количество включенных сегментов в i -ю наносекунду после начала измерения.

Формат выходных данных

В первой строке выведите k — количество возможных начальных значений счетчика по модулю 1 000 000 007. Затем выведите m различных значений числа на дисплее, подходящих под заданные измерения. В случае если реальное количество значений (до взятия по модулю 1 000 000 007) меньше чем m , выведите все подходящие значения. Выводите числа в произвольном порядке, каждое в отдельной строке.

Система оценки

Подзадача	Баллы	Дополнительные ограничения
1	17	$n = 1, m = 0, a_i \leq 100$
2	10	$n = 1, m \leq 10, a_i \leq 100$
3	19	$n \leq 100, a_i \leq 10$
4	22	$n \leq 100, a_i \leq 20$
5	32	нет дополнительных ограничений



Примеры

стандартный ввод	стандартный вывод
5 1 11 15 14 15 11	3 1151
10 1 13 10 14 12 13 9 12 11 10 11	1 102
4 10 29 28 29 29	108637 448765 812225 19541715 417773115 161177415 117347175 206215 8127315 12761415 54110175
1 6 6	6 111 41 14 6 77 9

Пояснения к примерам

В первом тесте, если дисплей изначально показывает число 1151, то он меняется следующим образом:

Число на дисплее	Число горящих сегментов
1151	$2 + 2 + 5 + 2 = 11$
1150	$2 + 2 + 5 + 6 = 15$
1149	$2 + 2 + 4 + 6 = 14$
1148	$2 + 2 + 4 + 7 = 15$
1147	$2 + 2 + 4 + 3 = 11$

Другие варианты для начального числа: 451 и 761, можно было вывести любое из них.



Задача D. Painting

Ограничение по времени: 1.5 секунд

У Мити есть прямоугольный клетчатый холст, который может быть представлен в виде таблицы размера $n \times m$, и k роботов. Пронумеруем строки таблицы от 1 до n сверху вниз, а столбцы — от 1 до m слева направо. Клетка (x, y) лежит на пересечении строки номер x и столбца номер y .

Изначально все клетки таблицы покрашены в белый цвет, который имеет номер 0. Робот номер i запрограммирован краской с цветом номер i ($1 \leq i \leq k$). Для каждого робота Митя выбрал подпрямоугольник холста, который задается четырьмя целыми числами $x_{i,1}$, $y_{i,1}$, $x_{i,2}$ и $y_{i,2}$ и содержит все клетки (x, y) , такие что $x_{i,1} \leq x \leq x_{i,2}$ и $y_{i,1} \leq y \leq y_{i,2}$ ($1 \leq x_{i,1} \leq x_{i,2} \leq n$, $1 \leq y_{i,1} \leq y_{i,2} \leq m$). После этого роботы друг за другом по одному разу в случайном порядке покрасили свой прямоугольник в свой цвет. При покраске робот изменяет цвет всех клеток на прямоугольнике на новый цвет. При этом старые цвета клеток полностью заменяются на новый и уже никак не могут быть восстановлены.

На следующий день холст увидел Леша. Он заметил, что в каждый цвет от 1 до k покрашена хотя бы одна клетка. И ему стало интересно, какие прямоугольники Митя назначил каждому роботу. А именно его интересует, мог ли Митя назначить каждому роботу прямоугольник таким образом, чтобы существовал порядок роботов, при котором получилась бы такая покраска холста. И если Митя мог выбрать такие прямоугольники, Лешу интересует, единственным ли образом он мог это сделать. Два способа считаются различными, если существует робот, которому в этих способах назначены различные прямоугольники. Два прямоугольника считаются различными, если у них не совпадает хотя бы одна из четырех координат x_1 , y_1 , x_2 или y_2 .

Формат входных данных

В первой строке даны три целых числа n , m и k — высота и ширина холста, количество роботов ($1 \leq n, m \leq 2000$; $1 \leq k \leq 1000$).

В следующих n строках даны по m целых чисел c_{ij} — цвета клеток таблицы, задающей холст ($0 \leq c_{ij} \leq k$).

Гарантируется, что для каждого цвета от 1 до k существует хотя бы одна клетка, покрашенная в него.

Формат выходных данных

Если не существует способа, которым Митя мог бы назначить роботам прямоугольники, чтобы существовал порядок роботов, при котором получилась бы такая покраска, выведите «No solution».

Если существует единственный способ, в первой строке выведите «Single solution». В следующих k строках выведите по четыре целых числа $x_{i,1}$, $y_{i,1}$, $x_{i,2}$ и $y_{i,2}$, задающие прямоугольник для i -го робота ($1 \leq x_{i,1} \leq x_{i,2} \leq n$, $1 \leq y_{i,1} \leq y_{i,2} \leq m$). И в последней строке выведите перестановку чисел от 1 до k — порядок работы роботов, при котором бы получилась такая раскраска.

Если способов больше одного, в первой строке выведите «Multiple solutions». В следующей строке выведите «First». В следующих строках выведите описание первого способа в том же формате, как и в предыдущем абзаце. В следующей строке выведите «Second». И в следующих строках выведите описание второго способа. Первый и второй способы должны различаться. Если существует больше двух различных способов, вы можете вывести любые два различных.

Система оценки

Подзадача	Баллы	Дополнительные ограничения
1	5	$k = 1$; $n, m \leq 300$
2	5	$k \leq 2$; $n, m \leq 300$
3	10	$k \leq 5$; $n, m \leq 300$
4	25	$n = 1$
5	25	$n, m, k \leq 300$
6	30	без дополнительных ограничений



Примеры

стандартный ввод	стандартный вывод
3 4 3 0 3 3 0 1 3 2 1 1 1 2 1	Single solution 2 1 3 4 2 3 3 3 1 2 2 3 1 3 2
2 2 2 1 2 2 1	No solution
2 1 2 2 1	Multiple solutions First 2 1 2 1 1 1 2 1 2 1 Second 1 1 2 1 1 1 1 1 1 2
2 3 3 0 3 0 2 0 1	Single solution 2 3 2 3 2 1 2 1 1 2 1 2 3 2 1



Задача E. Chicken Farm

Ограничение по времени: 1 секунда

Асхат — начинающий бизнесмен. Ему стало понятно, что программирование — бизнес неприбыльный, поэтому он решил создать куриную ферму.

На его ферме есть n цыплят, которые стоят в ряд. i -й цыпленок может съесть не больше a_i зерен. Есть m кормушек, каждая задается числами l_j, r_j и c_j : из j -й кормушки может есть i -й цыпленок, если $l_j \leq i \leq r_j$, причем из j -й кормушки всеми цыплятами суммарно может быть съедено не больше c_j зерен.

Как и в любом бизнесе, проблемы возникают буквально на пустом месте. Так и в этот раз к Асхату на ферму нагрянула СЭС. По их мнению, на ферме должны существовать как минимум два цыпленка, которые могут есть из всех кормушек. Иными словами, существует такое i , что $1 \leq i \leq n - 1$, и для всех кормушек выполняется $l_j \leq i$ и $i + 1 \leq r_j$. Все кормушки, которые не будут удовлетворять этому свойству, придется уничтожить. Асхат просит вас для каждого i найти, какое максимальное количество зерен можно будет скормить цыплятам, если оставить только кормушки, из которых могут есть цыплята i и $i + 1$.

Формат входных данных

Первая строка содержит целое число t — число тестов во входных данных ($1 \leq t \leq 2000$).

Далее следуют описания тестов. Первая строка каждого теста содержит два целых числа n и m — число цыплят и число кормушек, соответственно ($1 \leq n \leq 2000, 1 \leq m \leq 100\,000$). Следующая строка содержит n целых чисел a_i , число зерен, которое может съесть i -й цыпленок ($0 \leq a_i \leq 10^9$). Следующие m строк содержат по три целых числа l_j, r_j и c_j — описание j -й кормушки ($1 \leq l_j \leq r_j \leq n, 0 \leq c_j \leq 10^9$).

Сумма n по всем тестам не превосходит 2000.

Сумма m по всем тестам не превосходит 100 000.

Сумма $n \cdot m$ по всем тестам не превосходит 10^7 .

Формат выходных данных

Для каждого теста выведите $n - 1$ число: i -е из этих чисел должно равняться максимальному числу зерен, которое можно скормить цыплятам если оставить только кормушки, у которых $l_j \leq i$ и $r_j \geq i + 1$.

Система оценки

Подзадача	Баллы	$\sum n$	$\sum m$	Доп. ограничения
1	5	$\sum n \leq 100$	$\sum m \leq 100$	—
2	10	$\sum n \leq 500$	$\sum m \leq 500$	—
3	25	$\sum n \leq 1\,000$	$\sum m \leq 1\,000$	—
4	10	$\sum n \leq 2\,000$	$\sum m \leq 100\,000$	$l_i \leq l_{i+1}, r_i \leq r_{i+1}$
5	20	$\sum n \leq 500$	$\sum m \leq 100\,000$	—
6	30	$\sum n \leq 2\,000$	$\sum m \leq 100\,000$	—

Пример

стандартный ввод	стандартный вывод
1	4 4 9 4
5 3	
5 2 2 3 1	
1 4 4	
3 5 4	
3 4 1	



Пояснения к примерам

Если оставить кормушки, из которых могут есть цыплята с номерами 1 и 2, то останется только первая кормушка. В таком случае можно скормить первому цыпленку все зерна из нее, и всего будет съедено четыре зерна.

Аналогично получится, если оставить кормушки, из которых могут есть второй и третий цыплята.

Если оставить кормушки для цыплят с номерами 3 и 4, то все кормушки останутся. Тогда можно скормить первому цыпленку зерна из первой кормушки, а третьему и четвертому цыплятам — зерна из оставшихся кормушек. Таким образом получится девять съеденных зерен.

В последнем случае остаются кормушки, из которых могут есть четвертый и пятый цыплята. Останется только вторая кормушка, из которой цыплята съедят все зерна.