

Задача А. Рецепт мармелада

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Медвежонок Паддингтон очень любит есть мармелад. Однако, чтобы его съесть, мармелад надо сначала приготовить. У Паддингтона есть секретный рецепт своего любимого мармелада, состоящий из n пунктов a_i , причем каждый пункт описывается числом от 1 до m , означающим добавление одного из m ингредиентов.

Однако уже второй день Паддингтон не может найти свой рецепт, а мармелада уж очень как хочется. Поэтому он решил вспомнить про рецепт все что можно. Вот, что ему удалось вспомнить:

- Какие n и m использовались в рецепте;
- Ингредиенты в пунктах рецепта были расположены по неубыванию, а именно, для любых двух пунктов $i < j$ верно, что $a_i \leq a_j$;
- Последнее, что он смог восстановить в памяти — массив b_1, b_2, \dots, b_m , где b_i означает количество чисел в a , меньше либо равных i .

По этой информации Паддингтон хочет попытаться восстановить исходный рецепт. Однако, он не силен в программировании, и обратился с этой задачей к вам. Помогите ему!

Формат входных данных

В первой строке содержится два числа n и m — количество пунктов в рецепте и общее количество ингредиентов соответственно ($1 \leq n, m \leq 10^5$).

Во второй строке содержится m чисел b_i — массив, который смог восстановить в памяти Паддингтон ($0 \leq b_i \leq n$).

Формат выходных данных

В единственной строке через пробел выведите n чисел — исходный рецепт. Гарантируется, что ответ существует.

Система оценки

Эта задача состоит из двух подзадач. Для подзадач выполняются дополнительные ограничения, указанные в таблице ниже. Для получения баллов за подзадачу необходимо пройти все тесты данной подзадачи, а также все тесты всех предыдущих подзадач.

Обратите внимание, что **тест из условия** не подходит под ограничения некоторых подзадач, однако он обязательно **должен быть пройден** для того, чтобы решение было принято на проверку.

Подзадача	Баллы	Ограничения
1	39	В исходном рецепте все ингредиенты во всех пунктах различны
2	61	Полные ограничения

Примеры

стандартный ввод	стандартный вывод
3 2 1 3	1 2 2
5 6 1 2 3 4 5 5	1 2 3 4 5

Задача В. Стаканчики

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Генри Браун хочет провести приятный вечер за чтением газеты. К его несчастью, Паддингтон нашел огромную коробку пластиковых стаканчиков.

Мишка знает, что из стаканчиков можно строить красивые пирамиды, каждый стакан в которых, не считая основания, ставится на края четырех других. Таким образом, если в основании пирамиды находится прямоугольник, состоящий из $N \times M$ стаканов, то следующий уровень будет состоять из $(N - 1) \times (M - 1)$ стаканов, следующий из $(N - 2) \times (M - 2)$, и так далее... Уровни продолжают до тех пор, пока можно поставить стаканчик на края четырех других стаканчиков из предыдущего уровня.



Паддингтон никак не может определиться, какое основание будет у пирамиды, которую он построит, и сколько на нее уйдет стаканчиков. Паддингтону сложно посчитать такие числа, тем более, что они могут быть большими, поэтому он донимает мистера Брауна. Помогите Генри Брауну ответить на все вопросы настырного медвежонка.

Формат входных данных

В первой строке задано одно натуральное число t — число оснований, интересующих Паддингтона ($1 \leq t \leq 10^5$).

В следующих t строках заданы пары чисел N_i, M_i — размеры оснований ($1 \leq N_i, M_i \leq 10^9$).

Формат выходных данных

Для каждого основания в отдельной строке выведите число стаканчиков, которые потребуются, чтобы построить пирамиду из стаканчиков с соответствующим основанием.

Система оценки

Эта задача состоит из четырех подзадач. Для подзадач выполняются дополнительные ограничения, указанные в таблице ниже. Для получения баллов за подзадачу необходимо пройти все тесты данной подзадачи, а также все тесты всех необходимых подзадач. Необходимые подзадачи также указаны в таблице.

Обратите внимание, что **тест из условия** не подходит под ограничения некоторых подзадач, однако он обязательно **должен быть пройден** для того, чтобы решение было принято на проверку.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	30	$t, N_i, M_i \leq 100$	
2	30	$t \leq 10^5, N_i = M_i, N_i \leq 10^6$	
3	25	$t \leq 10^5, N_i \leq 10^6$	1, 2
4	15	$t \leq 10^5, N_i \leq 10^9$	1, 2, 3

Пример

стандартный ввод	стандартный вывод
3	1
1 1	14
2 5	14
3 3	

Задача С. Сеть дорог

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Лондон — очень старинный город. По ходу истории город разрастался, и в нем появлялись новые дороги. Сейчас дороги в Лондоне бывают двух типов: кольцевые и радиальные. Количество кольцевых дорог равно k . Пронумеруем их от 1 до k , тогда i -я кольцевая дорога представляет собой квадрат со сторонами параллельными осям координат, и с противоположными углами в точках $(-i, -i)$ и (i, i) . Любая радиальная дорога является отрезком и соединяет две кольцевые дороги с номерами, отличающимися ровно на 1, или кольцевую дорогу с номером 1 и точку $(0, 0)$. При этом, она не имеет других общих точек с кольцевыми дорогами, кроме своих концов. Никакие две радиальные дороги не пересекаются, но могут иметь общие концы.

Паддингтон хочет добраться из точки a с координатами (x_a, y_a) в точку b с координатами (x_b, y_b) , перемещаясь только по дорогам. Найдите минимальное расстояние, которое ему придется преодолеть.

Смотрите пояснение к тестам из примера для лучшего понимания условия.

Формат входных данных

В первой строке дано два целых числа n и k — количество радиальных дорог и количество кольцевых дорог ($0 \leq n \leq 10^5$, $1 \leq k \leq 10^9$). В следующих n строках дано по 4 целых числа x_{i1} , y_{i1} , x_{i2} и y_{i2} — координаты начала и конца i -й радиальной дороги ($-k \leq x_{i1}, y_{i1}, x_{i2}, y_{i2} \leq k$). В последней строке дано четыре целых числа x_a , y_a , x_b , y_b — координаты точек a и b , соответственно ($-10^9 \leq x_a, y_a, x_b, y_b \leq 10^9$).

Гарантируется, что дороги удовлетворяют ограничениям, описанным в условии.

Формат выходных данных

Выведите одно вещественное число с абсолютной или относительной погрешностью не более 10^{-6} — кратчайшее расстояние из точки a до точки b . Или -1 , если пути не существует.

Система оценки

Эта задача состоит из пяти подзадач. Для подзадач выполняются дополнительные ограничения, указанные в таблице ниже. Для получения баллов за подзадачу необходимо пройти все тесты данной подзадачи, а также все тесты всех необходимых подзадач. Необходимые подзадачи также указаны в таблице.

Обратите внимание, что **тест из условия** не подходит под ограничения некоторых подзадач, однако он обязательно **должен быть пройден** для того, чтобы решение было принято на проверку.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	12	$n = 0$, $k \leq 1\,000$	
2	22	$k \leq 30$, $n \leq 500$. Каждая радиальная дорога соединяет либо углы квадратов, либо центры сторон квадратов. Посмотрите пояснение к третьему тесту из примера для лучшего понимания.	
3	22	$k \leq 200$, $n \leq 500$	2
4	22	$k \leq 50\,000$	2, 3
5	22		2, 3, 4

Примеры

стандартный ввод	стандартный вывод
1 1 0 0 1 1 -1 -1 0 0	5.414213562373095
5 3 1 0 2 2 2 1 3 2 -1 1 0 2 -1 -2 0 -1 -3 -1 -2 -1 -3 -1 3 2	10.414213562373096
5 3 0 0 1 1 3 -3 2 -2 2 0 1 0 0 -1 0 0 -2 0 -3 0 1 3 0 0	14.828427124746192

Замечание

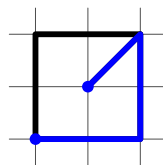


Рис. 1: Пояснение для первого теста

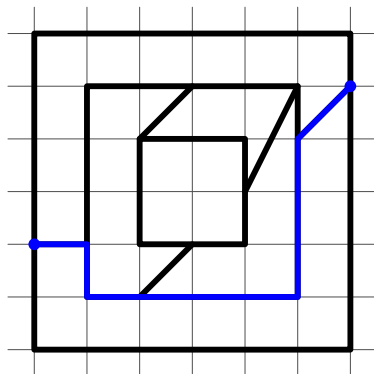


Рис. 2: Пояснение для второго теста

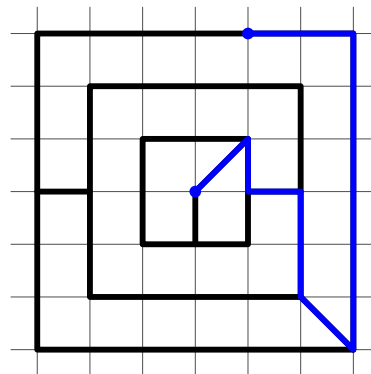


Рис. 3: Пояснение для третьего теста

Задача D. Равенство

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Для того, чтобы сосчитать, сколько мармелада понадобится сварить в каждый из дней, Паддингтону необходимо быть неплохо подкованным в арифметике, ведь заключенных в наше время очень много!

Сегодня Кастет дал Паддингтону несколько математических заданий. Каждое задание выглядит так: дана последовательность $a_1 a_2 \dots a_n$, состоящая из n цифр (иными словами, $0 \leq a_i \leq 9$ для всех i). Вместе с последовательностью Паддингтону дается два числа k и m .

Задача Паддингтона состоит в том, чтобы поставить между некоторыми парами соседних цифр знаки сложения, умножения или равенства, чтобы получилось верное выражение по модулю m . При этом требуется, чтобы знаков равенства было ровно k . Между некоторыми парами соседних цифр можно не ставить никаких знаков, в таком случае эти цифры «склеятся» в одно число.

Формально, Кастет будет проверять правильность выполнения задания так: сначала он объединит блоки цифр, идущих подряд, между которыми не стоит знаков равенства. При объединении цифр b_1, b_2, \dots, b_l получается число $b_1 \cdot 10^{l-1} + b_2 \cdot 10^{l-2} + \dots + b_{l-1} \cdot 10 + b_l$. После этого Кастет разобьет выражение на блоки, разделенные знаками равенства. Этих блоков должно быть ровно $k + 1$, иначе задание не будет зачтено. Затем, в каждом блоке будет подсчитано значение выражения. После этого Кастет проверит, что все получившиеся значения дают одинаковый остаток от деления на m , и в этом случае задание будет выполнено. Кастет проверяет задания лояльно, поэтому он разрешает Паддингтону использовать числа с ведущими нулями.

Задания бывают двух уровней сложности: в заданиях первого уровня Паддингтон может использовать только знаки сложения и равенства, а в заданиях второго — знаки сложения, равенства, а также умножения.

К сожалению, Паддингтон отвлекся на написание письма тете Люси, и не успел выполнить задания. Помогите ему справиться с ними!

Формат входных данных

Первая строка входных данных содержит единственное целое число q — количество заданий, которое необходимо выполнить ($1 \leq q \leq 1000$).

Далее следуют описания q заданий. Описание каждого задания состоит из двух строк.

Первая строка описания задания содержит четыре целых числа n, k, m, t — длина последовательности цифр в задании, количество знаков равенства, которые нужно использовать в ответе, число, остатки от которого будут сравниваться и уровень сложности задания, соответственно ($1 \leq k < n \leq 200, 1 \leq m \leq 1000, 1 \leq t \leq 2$). Если $t = 1$, то можно использовать только знаки сложения и равенства, а если $t = 2$, то можно использовать дополнительно знаки умножения.

Вторая строка описания задания содержит последовательность из n цифр a_i , записанных подряд ($0 \leq a_i \leq 9$).

Обратите внимание на ограничения, данные для подзадач.

Формат выходных данных

Для каждого задания, описанного во входном файле, в отдельной строке выведите ответ на него.

В случае, если у задания нет решения, выведите единственное слово «Fail» (без кавычек).

В случае, если решение есть, выведите исходную последовательность, поставив в нужных местах нужные знаки сложения, умножения и равенства. Никаких разделительных символов в выражении выводить не нужно. В случае, если решений несколько, выведите любое из них.

Для лучшего понимания формата входных и выходных данных изучите тест из примера.

Система оценки

Эта задача состоит из семи подзадач. Для подзадач выполняются дополнительные ограничения, указанные в таблице ниже. Для получения баллов за подзадачу необходимо пройти все тесты данной

подзадачи, а также все тесты всех необходимых подзадач. Необходимые подзадачи также указаны в таблице.

Обратите внимание, что **тест из условия** не подходит под ограничения некоторых подзадач, однако он обязательно **должен быть пройден** для того, чтобы решение было принято на проверку.

Обозначим за N суммарную длину всех заданий.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	11	$n \leq 4, k = 1, t = 1$	
2	11	$N \leq 10, m \leq 100, t = 1$	
3	11	$N \leq 10$	2
4	11	$N \leq 70, m \leq 500, t = 1$	2
5	16	$N \leq 200, m \leq 1000, t = 1$	2, 4
6	19	$N \leq 50, m \leq 100$	2
7	21	$N \leq 100, m \leq 100$	2, 6

Пример

стандартный ввод	стандартный вывод
8	Fail
5 2 3 1	3=3
98212	35+3=8
2 1 9 1	5=3+2
33	43=67=3
4 1 10 1	68*07=6
3538	Fail
3 1 12 2	711*6=6
532	
5 2 2 1	
43673	
5 1 10 2	
68076	
5 1 100 1	
38608	
5 1 4 2	
71166	