

Задача А. Азиатские страхи

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 256 Мб

Условие

Некоторое число, являющееся результатом вычисления факториала из n , назовем числом x . Число x считается счастливым, если в его разложении на простые множители простое число k встречается не кратное 4 количество раз, а также более двух раз.

Разложение на простые множители - это представление числа в виде произведения простых множителей, из которых оно состоит. Например, число 120 раскладывается на $2 * 2 * 2 * 3 * 5$.

Вам необходимо узнать, является ли x счастливым.

Формат входных данных

В первой строке находится целое число t - количество наборов входных данных. ($1 \leq t \leq 10^4$)

Далее для каждого набора входных данных в одной строке располагается два целых числа n и k . ($2 \leq n \leq 10^6$, $2 \leq k \leq n$)

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите "Yes", если число оказалось счастливым, или "No" в противном случае.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	5	Yes
	5 2	No
	6 2	No
	7 2	Yes
	8 2	No
	9 3	

Задача В. Большая IT компания

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 256 Мб

Условие

В большой IT компании сейчас ведется проект, который нужно закрыть в кратчайшие сроки. Для этого собрали большую команду IT специалистов, а именно:

- m_1 - разработчиков;
- m_2 - тестировщиков;
- m_3 - инженеров по развертыванию.

Для того чтобы закрыть проект, нужно разработать, протестировать и развернуть на сервере N сервисов. Так как проект хорошо разбили на подзадачи, время, которое требуется для разработки, тестирования и развертывания, для всех сервисов одинаковое.

Нужно помнить, что каждый специалист может работать только с одним сервисом в один момент времени. А также, специалист не может перейти к работе над следующим сервисом, не закончив работу с текущим. Чтобы не нарушать порядок разработки, сервис сначала нужно разработать, затем протестировать и только потом его можно начать развертывать.

Напишите программу, которая поможет большой IT компании, а именно найдет оптимальный порядок обработки сервисов и выведет минимальное для этого время.

Формат входных данных

В первой строке записано целое число N ($1 \leq N \leq 10^5$) - количество сервисов.

Во второй строке записано три целых числа m_1, m_2, m_3 ($1 \leq m_1, m_2, m_3 \leq 1000$) - количество разработчиков, тестировщиков и инженеров по развертыванию.

В третьей строке записано три целых числа t_1, t_2, t_3 ($1 \leq m_1, m_2, m_3 \leq 1000$) - время, которое требуется на разработку, тестирование и развертывание соответственно.

Формат выходных данных

Выведите одно целое число — минимальное количество минут, за которое можно разработать, протестировать и развернуть все сервисы.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 3 1 1 1 1 1	5

Задача C. CODE ровка

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 256 Мб

Условие

Отдыхая в баре с бесконечным числом математиков, у Вани родилась идея кодировки sws7 , не имеющей аналогов. В порыве вдохновения Ваня срочно закодировал все свои важные числа и довольным лёг спать. Однако на утро он обнаружил, что не может декодировать их обратно, поэтому очень ждёт вашей помощи!

Закодированный ряд представляет собой одно большое число x , не содержащее нулей. Чтобы вернуть исходный ряд нужно:

1. Разделить x на ряд чисел так, чтобы выполнялось условие $x_{i-1} < x_i$ и x_i было минимально возможным.

159 \rightarrow 1, 5, 9

1555 \rightarrow 1, 5, 55

Если в конце осталось число, меньше или равное, чем последнее добавленное, то это явно информационный шум и его нужно выбросить.

1232 \rightarrow 1, 2, 3

2. К каждому числу x_i из получившегося ряда применить формулу:

$x_i = a^{x_i} \bmod 7$, где a - это двузначное число, первая цифра которого - это первая цифра x_i , а вторая цифра - последняя цифра x_i .

Формат входных данных

В первой строке записано единственное положительное число x , которое не содержит нулей.

Формат выходных данных

Через пробел выведите получившийся ряд чисел.

Ограничения

$1 \leq \text{len}(x) \leq 170$

Пояснение к примеру

Во втором примере число разделится как 6 41 53:

$$66^6 \bmod 7 = 1$$

$$41^{41} \bmod 7 = 6$$

$$53^{53} \bmod 7 = 2$$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	1555	4 6 6
2	641539	1 6 2

Задача D. Детский утренник

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 3 сек
Ограничение памяти: 256 Мб

Условие

В некоторой школе для одаренных детей у ребят появился новый тренд - вставать в ряд по неубыванию роста, т.е так, чтобы рост следующего школьника в ряду был не меньше роста текущего. Однако быть самым низким не нравится никому, поэтому каждый из школьников говорит вам, что он не хочет стоять в последних a_i местах (некоторые школьники не капризные вовсе и могут стоять на последнем месте).

Всего есть n школьников, которые приходят в ряд по очереди. Вам требуется после каждого нового школьника сказать, ряд какой максимальной длины можно составить из них.

Формат входных данных

В первой строке входных данных находится целое число n . ($1 \leq n \leq 10^5$)

В следующих n строках располагается пара из вещественного и целого чисел h_i и a_i - рост школьника в сантиметрах и количество последних мест, на которые он не будет вставать. Рост имеет 6 знаков после точки. ($130 \leq h_i \leq 190$, $0 \leq a_i < n$)

Формат выходных данных

Вам требуется вывести n целых чисел: в i -ой строке максимальную длину ряда, который можно составить из первых i школьников. ($1 \leq i \leq n$)

Примеры тестов

№	Стандартный вход	Стандартный выход
1	5 174.500000 0 175.000000 1 173.000000 1 171.000000 0 180.000000 5	1 2 2 4 4
2	5 150.000000 1 150.000000 2 150.000000 0 150.000000 3 150.000000 4	0 0 3 4 5

Задача Е. Ежеминутные указания

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 2 сек
Ограничение памяти: 256 Мб

Условие

На выставке виртуальной реальности всем желающим дается возможность поиграть в VR-очках. Но так как участнику для игры нужно свободное пространство, то их расстановкой занимаются менеджеры. А также участник не может играть вечно, поэтому на игру ему отводится определенное время. Все хотят поиграть подольше, поэтому каждый участник будет стоять именно свои t_i минут.

Другими словами, i -ый участник встает в точку (x_i, y_i) , в которой он будет находиться t_i минут. Менеджеры ответственно выполняют свою работу, поэтому они не отправят человека в занятую точку.

Вы, как администратор площадки, обязаны отчитываться сколько человек находится в некотором прямоугольнике. Вы не занимаетесь общением с людьми, поэтому вам приходят указания от менеджеров вида:

- Поставить человека в точку (x_i, y_i)
- Сказать сколько человек находится на прямоугольнике, ограниченном точками (x_1, y_1) , (x_2, y_1) , (x_1, y_2) , (x_2, y_2) .

Прямоугольник описывается двумя точками (x_1, y_1) и (x_2, y_2) , где $x_1 \leq x_2$ и $y_1 \leq y_2$. Точка (x, y) лежит на прямоугольнике, если $(x_1 \leq x \leq x_2$ и $y_1 \leq y \leq y_2)$

На выполнение каждого указания вы тратите T минут. Когда вам приходит указание посчитать людей, вы распечатываете для себя расстановку в текущий момент времени и считаете людей по ней. А время игры участника начинается сразу после того, как вы поставите его в точку.

Формат входных данных

В первой строке содержится два целых числа N и T - количество указаний и время, которое вы тратите на выполнение указания. ($1 \leq N, T \leq 2 * 10^5$)

В следующих N строках описываются указания, первое число строки говорит о том, какое указание в этой строке:

- 1 - поставить человека в точку. Далее в этой строке следует 3 целых числа x_i, y_i, t_i . ($1 \leq x_i, y_i \leq 500, 1 \leq t_i \leq 10^4$)
- 2 - сказать сколько человек в прямоугольнике. Далее следует 4 целых числа: x_1, y_1, x_2, y_2 . ($1 \leq x_1, x_2, y_1, y_2 \leq 500$)

Всего K запросов вида 2.

Формат выходных данных

Вы должны вывести K натуральных чисел - ответы на запросы вида 2.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	7 5 1 1 1 10 2 0 0 5 5 2 0 0 5 5 1 2 2 10 1 2 1 5 2 0 0 5 5 2 0 0 5 5	1 1 2 0
2	13 1 1 1 1 100 1 2 1 100 1 3 1 100 1 4 1 100 1 5 1 100 1 1 5 100 1 2 2 100 1 3 3 100 2 1 1 5 5	8 3 2 1 0

2	1	2	5	5
2	1	3	3	5
2	1	5	1	5
2	5	5	5	5

Задача F. Фантастическое чёрно-красное дерево

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 512 Мб

Условие

Существует фантастическое черное-красное дерево - граф, который представляет из себя дерево (связный граф без циклов), в котором каждая вершина покрашена в один из двух цветов - красный или черный.

У вас есть не менее замечательная строка s , состоящая из K символов 'r' и 'b'. Ваша задача - выяснить сколько существует путей по дереву, таких, что если выписать цвета вершин, по которым вы прошли, то получится строка s .

Путь - последовательность вершин, которая ведет из одной вершины в другую и проходит по ребрам графа. Одна вершина встречается в пути не более одного раза.

Формат входных данных

В первой строке целое число N - количество вершин в графе.

Далее строка g состоящая из N символов 'r' и 'b', описывающая цвета вершин дерева.

Далее строка s , состоящая из K символов 'r' и 'b'.

В следующих строках два целых числа - a_i и b_i - начало и конец ребра.

Формат выходных данных

Одно целое число - количество различных путей.

Ограничения

$$2 \leq K \leq N \leq 10^4$$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	5 brbrr rbr 1 3 4 3 3 5 2 3	3
2	5 brrbr rb 1 2 1 5 3 4 5 4	4

Задача G. Говорят программисты много зарабатывают

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 256 Мб

Условие

Алексей, опытный программист, всегда был настороже по отношению к банкам и предпочитал самостоятельно контролировать свои доходы. Но скучать ему не приходилось: он решил разнообразить свою жизнь и сосредоточиться на красоте своей заработной платы.

Красивой зарплатой он называл ту, которая была задана простым числом и являлась палиндромом. Теперь Алексей хочет посчитать только такие зарплаты за последнее время. Помогите ему справиться с этим.

Формат входных данных

Первая строка входных данных содержит целое число n - количество зарплат Алексея.

Следующие n строк содержат целое число k зарплату Алексея за месяц в пуплях.

Формат выходных данных

Выведите единственное число - сумму только красивой зарплаты Алексея за n месяцев. Гарантируется, что сумма всех зарплат Алексея не превышает 10^9 пуплей.

Ограничения

$$1 \leq n \leq 1000$$

$$1 \leq k \leq 10^6$$

Примечание

Палиндром - число, одинаково читающееся в обоих направлениях.

Простое число - натуральное число, имеющее ровно два различных натуральных делителя.

Пример:

- 7 - простое число
- 101 - палиндром

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 1 7 11	18
2	2 191 8	191

Задача Н. Нерассудительность

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 256 Мб

Условие

У вас есть друг Петя, которому вы дали k рублей и отправили его в магазин купить круп, чтобы приготовить ужин. Вы сказали ему сколько и чего купить, но у него, конечно же, сразу всё вылетело из головы. Однако Петя - человек гордый, поэтому звонить и переспрашивать не будет. Он поступит нерассудительно - купит крупы наугад.

В магазине все крупы продаются только пакетами по 1кг и стоят одинаково.

Вы знаете об этой особенности Пети, и хотите рассчитать вероятность, что он купит именно то, что вы и попросили. Но для этого надо рассчитать сколько различных покупок может сделать Петя.

Две покупки являются различными, если существует такая крупа, что ее количество в первой покупке отличается от количества во второй покупке.

Петя отличается силой, в отличие от памяти, поэтому он может унести сколько угодно килограмм.

Петя может как купить на все деньги крупы одного вида, так и не купить круп вовсе (прийти в магазин просто так).

Формат входных данных

Во входной строке три целых числа: n , k и m - количество видов круп, количество денег у Пети и стоимость одного килограмма. ($1 \leq n \leq 10^5$, $1 \leq m \leq k \leq 10^5$)

Формат выходных данных

Выведите одно целое число - количество различных покупок, которые может сделать Петя.

Так как ответ может оказаться слишком большим, выведете его по модулю $10^9 + 7$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	1 2 1	3
2	2 4 2	6

Задача I. Игра в слова

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 1024 Мб

Условие

Миша и Паша играют в игру со словами. Игра достаточно трудная и правила у нее замудренные. Миша называет слово длины n , а Паша должен выбрать из ранее сказанных слов наиболее непохожее на только что сказанное.

Слова состоят из строчных латинских букв.

“Непохожесть” слова a на слово b определяется по следующей формуле:

$$\sum_{i=1}^n ((a_i - b_i) \cdot 100^{i-1}), \text{ если } a_i \geq b_i$$

$$\sum_{i=1}^n ((b_i - a_i + 50) \cdot 100^{i-1}), \text{ если } a_i < b_i$$

Где i - индекс буквы, a_i, b_i - алфавитные номера букв в словах на i -ом месте.

В нашем случае a - одно из ранее сказанных слов, b - только что сказанное слово.

В первый раунд Паше отвечать не нужно.

Пояснение

Непохожесть слова *code* на слово *work* вычисляется следующим образом:

порядок букв в алфавите:

$c - 3, o - 15, d - 4, e - 5$

$w - 23, o - 15, r - 18, k - 11$

$$(23 - 3 + 50)100^0 + (15 - 15)100^1 + (18 - 4 + 50)100^2 + (11 - 5 + 50)100^3 = 56640070$$

Формат входных данных

В первой строке располагается два целых числа n и k - длина слова и количество ходов в игре.

В следующих k строках располагаются строки длины n - слова, которые говорит Миша.

Формат выходных данных

Вам необходимо вывести $k - 1$ слов - ответы Паши.

Ограничения

$$1 \leq n \leq 6000 \quad 1 \leq k \leq 300$$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 5 abd cdc zdd ccd zxc	abd cdc cdc abd
2	4 4 code work tthe best	code work code

Задача J. Jade portal

Ввод / вывод: интерактивный

Ограничение времени: 1 сек

Ограничение памяти: 256 Мб

Условие

Это интерактивная задача. Параллельно с выполнением вашего решения жюри запускает проверяющую программу, с которой вы обмениваетесь сообщениями через стандартный ввод и вывод. Подробнее о протоколе взаимодействия написано ниже. Также в конце условия вы можете посмотреть корректные примеры взаимодействия с проверяющей программой на разных языках программирования.

Вы попали в странный мир, и у вас на руках оказалось n рун со странными знаками на них. Вы видите перед собой неработающий нефритовый портал, в котором есть k ячеек.

Немного поигравшись, вы заметили, что у рун есть своя энергия, и если попытаться засунуть руну в ячейку, то появится вспышка разных цветов:

- Белая, если в этой ячейке нет других рун.
- Синяя, если руна, которую вы пытаетесь положить имеет меньшую энергию, чем руны, которые уже лежат в этой ячейке.
- Зеленая, если руна, которую вы пытаетесь положить имеет такую же энергию, как руны, которые уже лежат в этой ячейке.
- Красная, если руна, которую вы пытаетесь положить имеет большую энергию, чем руны, которые уже лежат в этой ячейке.

Вы догадываетесь, что вам надо расположить руны по ячейкам в соответствии с их энергией. То есть в одной ячейке должны быть только руны одинаковой энергии.

Чтобы в ячейках не было рун разной энергии, когда вы видите синюю или красную вспышку, вы кладете только что добавленную руну обратно (К себе в карман или в ее прежнюю ячейку, в зависимости от того, где она была до этого). Неправильное расставление и возврат руны назад считаются за одну попытку.

Вам необходимо все руны раскидать по ячейкам. Гарантируется, что такое расположение рун существует. Времени у вас мало, поэтому вы можете сделать не более $15 * n$ попыток расставления.

Формат входных данных

При запуске решения на вход в первой строке подаются два целых числа k и n - количество ячеек в портале и количество рун.

Формат выходных данных

Вы можете делать запросы посредством вывода «? i j », где i, j - номер руны и номер ячейки, в которую вы собираетесь положить эту руну соответственно.

Ответом будет выведена одна из строчек «White», «Blue», «Green» или «Red» в зависимости от рун в ячейке.

Вы можете сделать не более $15 * n$ запросов. Вывод ответа не считается за запрос.

Вы можете вывести ответ, выведя «!», если считаете, что расположили все руны по их энергии.

Каждый вопрос и вывод ответа должен заканчиваться символом перевода строки $\backslash n$, а также необходимо выполнить сброс буфера:

Язык	C++	Pascal	Java	Python
Сброс буфера	cout.flush()	flush(output)	System.out.flush()	stdout.flush()

Ограничения

$$1 \leq k < n \leq 1000$$

Пояснение к примеру

Энергия рун в данном примере была следующая: $a_3 a_2 a_2 a_1$, где $a_3 > a_2 > a_1$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 4	? 1 1
	White	? 2 1

Blue	? 2 2
White	? 3 1
Blue	? 3 2
Green	? 4 2
Blue	? 4 3
White	!

Задача К. Кучка матриц

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 4 сек
Ограничение памяти: 256 Мб

Условие

У вас есть квадратная матрица A размера $n \times n$.

Также у вас есть k матриц B_k тоже размера $n \times n$.

От вас требуется найти такие две матрицы, произведение которых даст матрицу A (если таких матриц не существует, выведите -1).

Перемножение матриц:

Матрицы перемножаются по принципу строка на столбец. Результат кладётся в клетку их пересечения.

Рассмотрим это на примере матриц: $A_{2 \times 2} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ и $B_{2 \times 2} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$. $A \times B = C$.

$$C_{2 \times 2} = \begin{bmatrix} a_{11} * b_{11} + a_{12} * b_{21} & a_{11} * b_{12} + a_{12} * b_{22} \\ a_{21} * b_{11} + a_{22} * b_{21} & a_{21} * b_{12} + a_{22} * b_{22} \end{bmatrix}.$$

Общая формула выглядит так: $c_{ij} = a_{i1}b_{1j} + \dots + a_{in}b_{nj}$.

Формат входных данных

В первой строке находятся два целых числа k и n - количество матриц и их размер.

В следующих строках описываются матрицы B_k . Матрицы разделены пустыми строкам. Каждая строка матрицы располагается в новой строке. Элементы матрицы в строке разделены пробелами.

Далее описывается матрица A .

Формат выходных данных

Выведите -1, если не нашлась подходящая пара матриц.

Или два индекса найденных матриц в порядке возрастания в ином случае.

Ограничения

$$2 \leq k \leq 30$$

$$2 \leq n \leq 100$$

$$-10^3 \leq a_k, b_k \leq 10^3$$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 2 1 1 1 1 2 2 2 2 3 3 3 3 6 6 6 6	1 3
2	3 2 1 2 3 4 4 3 2 1 1 0	-1

0 1	
0 0	
0 0	

Задача L. Лабиринт токсиков

Входной файл: Стандартный вход
Выходной файл: Стандартный выход

Ограничение времени: 1 сек
Ограничение памяти: 256 Мб

Условие

Денис посетил вечеринку спортивных программистов в Точке кипения, где участники обсуждали различные алгоритмы. В ходе обсуждения возник спор о задачах на BFS. Одни считали их интересными, другие - душными. Через какой-то время спор почти дошел до драки между участниками, после чего любители задач на BFS назвали своих противников "токсиками" и решили закрыть их в Точке кипения. В свою очередь Точка кипения представляет собой лабиринт.

Точка кипения представлена матрицей размерностью $n \times m$. Она может содержать следующие символы:

- '.' - пустая клетка;
- '#' - стена;
- 'S' - любитель BFS;
- 'T' - токсик.

Выход из точки кипения один и находится в правом нижнем углу лабиринта, то есть имеет координаты (n, m) .

Любители BFS хотят заблокировать противников, а именно заменить некоторые пустые клетки на стены так, чтобы все любители BFS смогли добраться до выхода, а все токсики остались заперты в Точке кипения.

Клетки, которые изначально содержат символы 'S' или 'T', не могут быть заблокированы, но через них можно проходить.

Помогите Денису определить, можно ли заблокировать некоторые пустые клетки в Точке кипения так, чтобы **ВСЕ** любители BFS смогли выйти, но при этом токсики бы не смогли. Учтите, что разрешается не блокировать клетки вовсе и блокировать выход из точки кипения. Гарантируется, что выход из лабиринта всегда представляет собой пустую клетку.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) - количество наборов входных данных.

Для каждого набора входных данных:

- первая строка содержит два целых числа n и m ($1 \leq n, m \leq 100$) - размерность лабиринта;
- в каждой из следующих n строк записано m символов. Символ '.' обозначает пустую клетку, '#' - стену, 'S' - любителя BFS, а 'T' - токсика.

Формат выходных данных

Для каждого набора входных данных выведите "YES" или "NO", в зависимости от того, можно ли заменить некоторые пустые клетки на стены, чтобы удовлетворить описанным ограничениям.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	2 3 3 S.. ... T.. 2 2 ST T.	YES NO