

Задача А. Корневой ним

Имя входного файла: `sqrtnim.in`
Имя выходного файла: `sqrtnim.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Для игры в корневой ним используются следующие правила. Перед двумя игроками лежит кучка из n камней. Они по очереди забирают оттуда камни. Если в кучке сейчас лежат k камней, то игрок может взять из неё от 1 до $\lfloor \sqrt{k} \rfloor$ камней, включительно. Например, из кучки из 10 камней можно брать 1, 2 или 3 камня. Проигрывает игрок, который не может сделать ход.

По заданному n определите, победит ли первый игрок при правильной игре обоих игроков.

Формат входных данных

Входной файл содержит единственное число n ($1 \leq n \leq 10^{12}$) — количество камней в кучке.

Формат выходных данных

Выведите WIN в случае победы первого игрока, и LOSE, если ему победить не удастся.

Примеры

<code>sqrtnim.in</code>	<code>sqrtnim.out</code>
3	WIN
5	LOSE

Задача В. Paper2. Листок бумаги

Имя входного файла: paper2.in
Имя выходного файла: paper2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Андрей и Никита на скучном зачете в ЛКШ решили поиграть в интересную игру. Они берут прямоугольный листок клетчатой бумаги размером $N \times M$. После этого они по очереди делают следующее — каждый выбирает некоторую незачеркнутую клетку и вычеркивает все клетки, находящиеся с ней в одном столбце и строке до ближайшей зачеркнутой клетки либо до границы. Проигрывает тот, кто не может сделать ход. Андрей ходит первым.

Определите, может ли Андрей выиграть при оптимальной игре Никиты.

Формат входных данных

Единственная строка содержит два натуральных числа N и M ($1 \leq N, M \leq 60$) — размеры прямоугольника.

Формат выходных данных

В первой строке выведите 'Win.', если Андрей может выиграть, и 'Loss.' в противном случае.

Если выигрыш возможен, во второй строке выведите описание одного из выигрышных ходов: координаты клетки, которую нужно выбрать первой.

Примеры

paper2.in	paper2.out
3 3	Win. 3 3
2 2	Loss.

Задача С. Максимальное значение функции Гранди

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Какое максимальное значение может принимать функция Гранди в графе с m ребрами?

Формат входных данных

`aplusb.in` или стандартный ввод

Формат выходных данных

`aplusb.out` или стандартный вывод

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
------------------------	-------------------------

Задача D. Bricks. Кирпичная стена

Имя входного файла: `bricks.in`
Имя выходного файла: `bricks.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Мы хотим построить стену высоты N , длины M и толщины 1. У нас есть бесконечное количество кирпичей $1 \times 1 \times 2$ и $1 \times 1 \times 3$. Кирпичи можно класть только горизонтально, т.е. длинная сторона должна идти вдоль длины стены.

Рассмотрим общие вертикальные стороны кирпичей (т.е. находящиеся внутри стены). Для прочности стены должно выполняться следующее: ни для каких двух соседних слоев кирпичей не должно существовать общей вертикальной границы, проходящей через оба слоя. Например, такая стена:

2	2
2	2

не подходит под требования, поскольку граница посередине проходит через оба слоя.

Найдите количество вариантов выстроить стену, удовлетворяющую требованиям, по модулю $10^9 + 7$.

Формат входных данных

В первой строке записаны числа N и M ($1 \leq N \leq 10^{18}$, $1 \leq M \leq 16$).

Формат выходных данных

Выведите ответ на задачу по модулю $10^9 + 7$.

Примеры

<code>bricks.in</code>	<code>bricks.out</code>
1 5	2
2 4	0

Задача Е. Количество слов

Имя входного файла: numwords.in
Имя выходного файла: numwords.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дан детерминированный конечный автомат. Определите, сколько существует различных слов длины K , принимаемых данным автоматом.

Формат входных данных

Первая строка входных данных содержит два целых числа N и M ($1 \leq N \leq 30$) — количество состояний в автомате и количество переходов. В следующих M строках записаны переходы данного автомата. Каждый переход задается тройкой S_i, C_i, T_i , где S_i — номер исходного состояния перехода ($1 \leq S_i \leq N$), C_i — символ, по которому осуществляется переход (строчная буква латинского алфавита), T_i — конечное состояние перехода ($1 \leq T_i \leq N$).

Далее записано число T — количество терминальных состояний автомата ($0 \leq T \leq N$). В следующей строке записано T различных чисел — номера терминальных состояний. Последняя строка входных данных содержит число K ($0 \leq K \leq 10^9$) — длина входного слова.

Начальное состояние автомата имеет номер 1. Если в процессе работы автомата появится невозможный переход (то есть возникает комбинация состояния и символа, не описанная в списке возможных переходов), то такое входное слово считается не распознанным автоматом.

Формат выходных данных

Выведите остаток от деления числа всевозможных входных слов длины K , распознаваемых данным автоматом, на $10^9 + 7$.

Примеры

numwords.in	numwords.out
5 8 1 a 2 1 b 3 2 a 4 2 b 3 4 b 3 3 a 2 3 b 5 5 a 2 4 2 3 4 5 4	10
1 3 1 a 1 1 b 1 1 c 1 1 1 1000000000	235939645

Задача F. Последовательности из 0 и 1 без k одинаковых символов подряд

Имя входного файла: k-equal.in
Имя выходного файла: k-equal.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По данным натуральным n и k определите количество последовательностей длины n из 0 и 1, не содержащих k одинаковых символов подряд.

Формат входных данных

В первой и единственной строке заданы числа n и k . $n \leq 10^9$, $2 \leq k \leq n + 1$, $k \leq 31$. Ответ выведите по модулю $10^9 + 7$.

Формат выходных данных

Выведите ответ по модулю $10^9 + 7$

Примеры

k-equal.in	k-equal.out
3 3	6
100 2	2

Задача G. Сложность Independent Set

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Все мы знаем, что задача INDEPENDENT SET NP-трудная. Докажите это, сведя к ней каноничную NP-трудную задачу 3-SAT. А именно, по данной 3-КНФ формуле выведите граф и целое число k , такие что данный граф имеет независимое множество размера k тогда и только тогда, когда входная формула имеет выполняющее означивание.

Формат входных данных

В первой строке даны два целых числа n и m ($1 \leq n, m \leq 100\,000$) — число переменных и кловов соответственно. В следующих m строках перечислены кловы, описание i -ого клова начинается с числа k_i ($1 \leq k_i \leq 3$) — количества переменных в клове. Следующие за ним k_i чисел задают литералы в i -ом клове: положительное число j соответствует переменной x_j , отрицательное число $-j$ — отрицанию переменной \bar{x}_j .

Формат выходных данных

В первой строке входного файла выведите 3 целых числа n, m и k ($1 \leq n, m \leq 1\,000\,000, 1 \leq k \leq n$) — число вершин и рёбер в графе, а также размер независимого множества.

В следующих m строках выведите рёбра графа по одному в строке.

Примеры

стандартный ввод	стандартный вывод
3 2 3 1 2 3 3 -1 -2 3	12 15 5 1 4 2 5 3 6 7 1 7 8 8 2 8 9 9 3 9 7 10 4 10 11 11 5 11 12 12 3 12 10
2 2 1 1 1 -1	6 6 4 1 3 2 4 5 1 5 5 6 3 6 6

Задача Н. Трудность Set Cover

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Покажите, что задача SET COVER является NP-трудной.

В задаче SET COVER вам дан универс (просто множество) из n элементов и множество его подмножеств, и требуется проверить, можно ли выбрать не более k подмножеств, целиком покрывающих универс

Формат входных данных

`aplusb.in` или стандартный ввод

Формат выходных данных

`aplusb.out` или стандартный вывод

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
------------------------	-------------------------

Задача I. Coloring и SAT

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Постройте полиномиальное сведение задачи 3-COLORING к задаче 3-SAT.

Формат входных данных

`aplusb.in` или стандартный ввод

Формат выходных данных

`aplusb.out` или стандартный вывод

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
------------------------	-------------------------

Задача J. Точка в многограннике

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Дан выпуклый многогранник и точка. Проверьте принадлежность точки многограннику.

Формат входных данных

`aplusb.in` или стандартный ввод

Формат выходных данных

`aplusb.out` или стандартный вывод

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
------------------------	-------------------------

Задача К. Пересечение двух сфер

Имя входного файла: `spheres.in`
Имя выходного файла: `spheres.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В пространстве даны две сферы, не совпадающие и пересекающиеся по какой-то окружности. Найдите уравнение плоскости, в которой лежит данная окружность.

Формат входных данных

В первой строке входных данных записаны четыре числа: координаты центра и радиус первой сферы, во второй строке записаны четыре числа: координаты центра и радиус второй сферы. Все числа целые, не превосходят по модулю 10^6 , значения радиусов положительные. Гарантируются, что сферы не совпадают и пересекаются по окружности.

Формат выходных данных

Программа должна вывести четыре действительных числа: коэффициенты A, B, C, D уравнения плоскости. Выводите коэффициенты с максимально возможной точностью.

Примеры

<code>spheres.in</code>	<code>spheres.out</code>
2 -1 3 3 1 1 -1 4	-1.0 2.0 -4.0 9.0

Задача L. Взаимное расположение плоскостей

Имя входного файла: `planes.in`
Имя выходного файла: `planes.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Даны две плоскости, заданные уравнениями $A_1x + B_1y + C_1z + D_1 = 0$ и $A_2x + B_2y + C_2z + D_2 = 0$. Определите взаимное расположение этих плоскостей в пространстве (совпадают, параллельны или пересекаются).

Формат входных данных

Программа получает на вход восемь целых чисел, не превосходящих по модулю 1000: коэффициенты уравнения первой плоскости A_1, B_1, C_1, D_1 , затем коэффициенты уравнения второй плоскости A_2, B_2, C_2, D_2 . Числа A_1, B_1, C_1 не равны нулю одновременно, числа A_2, B_2, C_2 не равны нулю одновременно.

Формат выходных данных

Если плоскости совпадают, выведите число 2, если плоскости пересекаются, выведите число 1, если плоскости параллельны, выведите число 0.

Примеры

<code>planes.in</code>	<code>planes.out</code>
1 2 3 4 4 3 2 1	1
1 2 3 4 1 2 3 5	0

Задача М. Теорема Вильсона

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Докажите, что натуральное число p является простым тогда и только тогда когда $(p - 1)! + 1$ делится на p

Формат входных данных

`aplusb.in` или стандартный ввод

Формат выходных данных

`aplusb.out` или стандартный вывод

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
------------------------	-------------------------

Задача N. Система линейных сравнений

Имя входного файла: `chinese.in`
Имя выходного файла: `chinese.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дана система из двух линейных сравнений:

$$\begin{cases} x \equiv a \pmod{n}, \\ x \equiv b \pmod{m}; \end{cases}$$

где числа n и m не обязательно взаимно простые. Решите эту систему или определите, что она не имеет решений.

Формат входных данных

В первой строке входного файла записано единственное число $1 \leq t \leq 100\,000$. В следующих t строках содержатся по четыре целых числа a, b, n, m , задающих одну систему сравнений. Все числа не превосходят по модулю 10^4 , $n > 1$, $m > 1$.

Формат выходных данных

Программа должна вывести t строк, по одной на каждую систему.

В случае, если система не имеет решений, выведите строку "NO".

В случае, если решение есть, то необходимо вывести слово "YES" и два таких числа x_0 и p , $0 \leq x_0 < p$, такие, что множество чисел $x = x_0 + kp$, где k — произвольное целое число является решением данной системы.

Примеры

<code>chinese.in</code>	<code>chinese.out</code>
3	YES 38 45
3 2 5 9	YES 1 45
1 1 5 9	NO
7 13 20 24	

Задача О. Граничные точки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На плоскости заданы координаты вершин многоугольника в порядке их обхода. Вершины имеют целые координаты. Многоугольник не обязательно выпуклый. Требуется найти количество точек с целыми координатами, которые лежат на его границе.

Формат входных данных

Сначала записано число N — количество вершин многоугольника ($3 \leq N \leq 100$), затем N пар целых чисел, задающих координаты его вершин. Все координаты не превосходят 10^4 по абсолютному значению.

Формат выходных данных

Выведите количество точек, которые лежат на границе этого многоугольника.

Примеры

стандартный ввод	стандартный вывод
4 0 2 2 2 2 0 0 0	8

Задача Р. Матожидание сна

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

Саша бросает монетку раз в минуту. Саша хочет спать. Поэтому она решила, что пойдет спать, после того, как решка выпадет 31 раз. Найдите матожидание количества минут, через которое Саша пойдет спать.

Формат входных данных

`aplusb.in` или стандартный ввод

Формат выходных данных

`aplusb.out` или стандартный вывод

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
------------------------	-------------------------

Задача Q. Перемешивание и XOR

Имя входного файла: `permxor.in`
Имя выходного файла: `permxor.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Вася хочет сгенерировать случайный тест для одной задачи. Для этого он выбрал последовательность из n чисел x_i и перестановку p_i . Далее он взял последовательность a_i из n чисел и m раз проделал с ней следующую операцию.

Подбросим монетку:

- если выпадет решка, применим к последовательности x перестановку p (то есть, сделаем $x[i] := x[p[i]]$),
- если выпадет орел — поксорим a с x (то есть, сделаем $a[i] := a[i] \oplus x[i]$)

Посчитайте математическое ожидание суммы всех элементов последовательности a после совершения всех операций.

Формат входных данных

В первой строке заданы числа n ($1 \leq n \leq 1000$) и m ($1 \leq m \leq 1000$). Во второй строке задана перестановка p_i , в третьей строке — последовательность x_i ($0 \leq x_i \leq 10^9$).

Формат выходных данных

Выведите одно число — математическое ожидание суммы всех элементов последовательности a после совершения всех операций.

Примеры

<code>permxor.in</code>	<code>permxor.out</code>
2 2 2 1 2 5	3.5

Задача R. Красно-синее перемешивание

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У нас есть n карточек, пронумерованных числами $1, \dots, n$. На карточке i написана красная цифра r_i и синяя цифра b_i .

Мы переупорядочиваем все n карточек слева направо в случайном порядке, так что все перестановки $1, \dots, n$ равновероятны. Затем мы читаем все красные цифры слева направо, и получаем целое число R . Аналогично мы читаем все синие цифры и получаем число B . Ведущие нули в числах игнорируются. Если все цифры в числе равны нулю, то такое число равно 0.

Два игрока, Красный и Синий, заключают пари. Красный ставит на то, что после перемешивания $R > B$, а Синий ставит на то, что $R < B$. Если в итоге $R = B$, то объявляется ничья и никто не выигрывает спор.

Определите, кто из двух игроков имеет лучшие шансы выиграть спор, либо что шансы игроков одинаковы.

Формат входных данных

В первой строке записано одно целое число T ($1 \leq T \leq 100$) — количество тестовых примеров.

Далее следуют описания T примеров. Описание каждого примера начинается со строки с одним целым числом n ($1 \leq n \leq 1000$) — количеством карточек.

В следующей строке записано n цифр r_1, \dots, r_n — красные цифры на карточках $1, \dots, n$ соответственно.

В следующей строке записано n цифр b_1, \dots, b_n — синие цифры на карточках $1, \dots, n$ соответственно.

Цифры в каждой строке записаны без разделителей.

Формат выходных данных

Выведите T ответов на примеры по порядку, по одному ответу на строку.

Если шансы Красного игрока на победу строго выше, выведите «RED».

Если шансы Синего игрока на победу строго выше, выведите «BLUE».

Если шансы игроков совпадают, выведите «EQUAL».

Обратите внимание, что регистр всех ответов важен.

Пример

стандартный ввод	стандартный вывод
3	RED
3	BLUE
777	EQUAL
111	
3	
314	
159	
5	
09281	
09281	

Задача S. Регулярные подстроки

Имя входного файла: `aplusb.in`
Имя выходного файла: `aplusb.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

L — регулярный язык. Докажите, что язык подстрок слов из L тоже регулярный.

Формат входных данных

`aplusb.in` или стандартный ввод

Формат выходных данных

`aplusb.out` или стандартный вывод

Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
------------------------	-------------------------

Задача Т. ДКА для чисел кратных k

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

Постройте детерминированный конечный автомат над алфавитом $\{0, 1\}$, допускающий только строки, являющиеся двоичным представлением чисел, делящихся на k и не содержащих ведущих нулей.

Формат входных данных

В единственной строке входного файла дано число k ($1 \leq k \leq 31$).

Формат выходных данных

В первой строке выходного файла выведите число N — количество состояний автомата ($1 \leq N \leq 100$).

Состояния нумеруются целыми числами от 1 до N .

В следующей строке выведите число s ($1 \leq s \leq N$) — номер начального состояния, затем число T — количество терминальных состояний, затем T чисел от 1 до N — номера терминальных состояний.

В следующих N строках выведите описание функции переходов. (В i -й из этих N строк выведите два числа — номера состояний, в которые из состояния с номером i ведут переходы по символам 0 и 1 соответственно).

Примеры

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