

Задача А. Интерактивные смайлики

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

Это интерактивная задача. Но что самое главное – это задача про смайлики :)

Как известно, смайлики бывают двух типов: веселые и грустные. Каждый день суперкомпьютер Lesli загадывает новый смайлик. Программисты пытаются угадать, какой смайлик загадала Lesli, веселый или грустный. Помогите программистам научиться безошибочно угадывать тип смайлика, который загадала Lesli.

Lesli загадывает смайлик на доске размера $10^9 \times 10^9$. Вся доска заполнена ноликами, кроме тех клеток, в которых расположен загаданный смайлик. Сам смайлик состоит из основы размера 41×41 , которая приведена на третьей странице условия. Координаты клеток доски по каждому измерению нумеруются от 1 до 10^9 . Ось x направлена слева-направо, а ось y снизу-вверх, как принято в Декартовой системе координат.

Так как Lesli не просто компьютер, а суперкомпьютер, она может основу смайлика увеличивать в k раз, где k – целое положительное нечетное число. После увеличения в k раз основа смайлика имеет размер $41 \cdot k \times 41 \cdot k$. Каждому квадрату 1×1 исходной основы соответствует квадрат $k \times k$ увеличенной основы, если в исходном квадрате стояло число 0, то в соответствующем квадрате $k \times k$ увеличенной основы все числа будут равны 0, если в исходном квадрате стояло число 1, то в соответствующем квадрате $k \times k$ увеличенной основы все числа будут равны 1.

После того, как Lesli увеличивает основу смайлика в k раз, она рисует его на доске размера $10^9 \times 10^9$. Гарантируется, что смайлик целиком помещается на доске. Более того, гарантируется, что в первом и последнем столбце, а также в первой и последней строке таблицы нет ни одного квадрата в котором записано число 1.

Отметим, что у основы смайлика всегда есть центральный квадрат, так как длина стороны основы смайлика всегда нечетна, в том числе после увеличения в нечетное число раз k .

После того, как основа смайлика нарисована Lesli рисует ему веселую или грустную улыбку. Форма улыбки показана на третьей странице условия, под изображением основы смайлика. Слева изображена веселая улыбка, а справа – грустная. Улыбку, также как и основу смайлика, Lesli может увеличить в m раз, где m – целое положительное число любой четности.

После того, как Lesli увеличивает улыбку смайлика в m раз, она рисует ее на основе смайлика. Место для улыбки выбирается таким образом, чтобы все квадраты, которые накрыла улыбка, содержали число 1. Когда улыбка нарисована, во всех накрытых ею квадратах число 1 заменяется на 0. Улыбку Lesli рисует таким образом, чтобы все ее квадраты находились ниже, чем центральный квадрат основы смайлика. Кроме того, улыбка рисуется таким образом, чтобы не задеть крайние квадраты основы. Крайние квадраты основы, это такие квадраты в которых написано число 1 и это либо самая нижняя, либо самая верхняя единица в столбце, либо самая правая, либо самая левая единица в строке.

Пример веселого смайлика с параметрами $k = 1, m = 2$ можно посмотреть на четвертой странице условия. Пример грустного смайлика с параметрами $k = 1, m = 1$ можно посмотреть на пятой странице условия.

Чтобы угадать тип смайлика разрешается сделать не более 500 запросов. Каждый запрос должен задавать некоторый прямоугольник, содержащийся целиком на доске размера $10^9 \times 10^9$. Ответ на запрос – это количество единичек внутри заданного прямоугольника.

Формат входных данных

После каждого запроса требуется считать одно число из входных данных (стандартный ввод). Эта число равно количеству единичек в запрашиваемом прямоугольнике.

Формат выходных данных

Не более 500 раз вы можете сделать запрос – вывести через пробел в одной строке четыре целых числа $x_1, y_1, x_2, y_2, 1 \leq x_1 \leq x_2 \leq 10^9, 1 \leq y_1 \leq y_2 \leq 10^9$, – координаты задающие расположение

прямоугольника. После вывода координат необходимо вывести перевод строки и сделать операцию “flush”. После сброса буфера необходимо считать ответ на запрос из входных данных.

В любой момент можно перестать делать запросы и вывести число -1, а затем в новой строке вывести “HAPPY” без кавычек, если нарисованный смайлик веселый, или “SAD” без кавычек, если нарисованный смайлик грустный. После этого требуется сделать “flush” и завершить программу.

Для сброса буфера вывода (то есть для операции “flush”) сразу после вывода нужно сделать:

fflush(stdout) в языке C++

System.out.flush() в языке Java

stdout.flush() в языке Python

flush(output) в языке Pascal

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 0 0 0 0 | 0 |
| 0 0 0 1 | 0 |
| 0 1 4 4 | 3 |
| -1 | |
| HAPPY | |

Задача В. Расписание докладов

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 4 секунды |
| Ограничение по памяти: | 256 мегабайт |

Алексей – организатор конференции по машинному зрению. Конференция будет проходить на протяжении n полных недель, и на ней выступят k докладчиков. Каждый докладчик подготовил три выступления. Алексей сейчас составляет расписание докладов, которое представляет из себя таблицу $7 \times n$. Расписание нужно составить таким образом, чтобы у каждого докладчика из трех его выступлений нашлись два выступления с разницей в 7 дней, и два выступления с разницей в 1 день. Также, для удобства администрирования, требуется, чтобы три дня выступлений каждого докладчика образовывали в таблице расписания связную область.

Напомним, что подмножество клеток таблицы образует связную область в том случае, если из любой клетки подмножества можно добраться до любой другой клетки подмножества за конечное число переходов, при условии, что переход от одной клетки подмножества к другой клетке подмножества можно делать только если у них есть общая сторона.

Алексей хорошо разбирается в математике и поэтому смог вычислить, что $7n - 3k$ дней конференции будут свободны от выступлений, эти дни будут выходными. Но и тут не все так просто. У Алексея есть список из m дней, в которые выходные ставить нельзя. Теперь Алексей хочет сначала перебрать все корректные способы расположить выходные дни в таблице $7 \times n$. Затем для каждого такого способа он хочет посчитать количество способов распределить оставшиеся $3k$ дней на k троек, таким образом, чтобы для каждой тройки выполнялись изложенные выше требования (порядок троек в разбиении не важен). После этого Алексей хочет просуммировать все получившиеся результаты.

Помогите Алексею с этими расчетами. Так как ответ может быть слишком большим, его нужно вывести по модулю $10^9 + 33$.

Формат входных данных

В первой строке через пробел даны два целых числа n и k , $2 \leq n \leq 100$, $7n - 20 \leq 3k \leq 7n$.

Во второй строке дано целое число m , $0 \leq m \leq 7n$.

В следующих m строках через пробел даны по два целых числа w_i и d_i , задающие день в который нельзя ставить выходной, где w_i – это номер недели, а d_i – номер дня недели, $1 \leq w_i \leq n$, $1 \leq d_i \leq 7$.

Формат выходных данных

В единственной строке выведите одно целое число – ответ на задачу по модулю $10^9 + 33$.

Пример

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

Задача С. Дети и Снежинки

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 2.5 секунд |
| Ограничение по памяти: | 128 мегабайт |

Зимним утром пятая группа детского сада “Солнышко” играет на дорожке во дворе. Дорожка имеет длину n метров и состоит из клеток с номерами от 1 до n . Клетка имеет размер 1 метр.

Дети занимают некоторые клетки дорожки. При этом, на одной и той же клетке могут стоять два и более ребёнка. Игра заключается в том, чтобы уклониться от снежинок, которые падают на дорожку. Все дети в группе имеют одинаковый рост h . За один ход каждый ребёнок может перебежать на соседнюю клетку вправо или влево или остаться на той же клетке.

Так как в данной задаче время дискретно, то для определённости примем, что в начале каждого хода все дети перебегают на некоторые клетки, и только после этого снежинки опускаются на 1 метр.

Каждая снежинка падает строго на определённую клетку дорожки. За один ход снежинка падает вниз на 1 метр. Когда снежинка, падающая на клетку i достигает высоты h , все дети, находящиеся на этой клетке выходят из игры. Если ребёнок прибегает на клетку на которой расположена снежинка на высоте h или ниже, то он выбывает из игры. Как только снежинка достигла высоты 0, клетку снова можно занимать. Известно, что в начале игры все снежинки находятся выше детей.

Воспитательница Тамара хочет организовать игру так, чтобы к концу m ходов на дорожке осталась хотя бы один ребёнок. Вам даны исходные позиции всех k детей и начальные положения всех l снежинок. Помогите воспитательнице, определите какое максимальное количество детей может остаться на дорожке после m ходов.

Формат входных данных

В первой строке через пробел даны пять целых чисел: n – количество клеток на дорожке, m – количество ходов, k – количество детей, l – количество снежинок, h – рост детей, $1 \leq n, k, l, h \leq 10^6$, $1 \leq m \leq 10^3$.

В следующей строке через пробел даны k целых чисел a_1, a_2, \dots, a_k – начальные позиции детей, $1 \leq a_i \leq n$.

В следующих l строках через пробел даны по два целых числа t_i и h_i – номер клетки и начальная высота снежинки, $1 \leq t_i \leq n$, $h + 1 \leq h_i \leq 10^6$.

Формат выходных данных

В единственной строке выведите одно целое число – ответ на задачу.

Примеры

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 3 3 2 2 2 1 2 1 3 2 3 | 1 |
| 6 10 7 7 3 1 2 2 3 4 5 6 6 10 2 4 2 6 3 5 4 4 5 6 6 10 | 6 |

Задача D. Много смайликов

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

«Опять эти задачи про смайлики!» – грустил Серёжа на олимпиаде. Действительно, на этот раз авторы дали бесконечное число задач, пронумерованных натуральными числами $(1, 2, 3, \dots)$, и все они были про смайлики. Серёжа много тренировался перед олимпиадой, и выбрал себе лучшую тактику: после задачи с номером x он решает задачу с номером $x \text{ xor } (x/2)$, где xor – это побитовое исключающее или, а деление производится с округлением вниз. Например, $4 \text{ xor } 8$ равно 12, $7 \text{ xor } 11$ тоже равно 12, $5 \text{ xor } (5/2)$ равно 7.

Серёжа считает задачу с номером x хорошей, если он решит k задач (начиная с x , выбирая их по своей тактике, при этом, возможно он решит некоторые задачи не по одному разу), а $(k + 1)$ -й задачей опять окажется x . Помогите Серёже – для данного k найдите количество хороших задач. Так как ответ может быть большим, выведите его по модулю $10^9 + 7$.

Формат входных данных

В единственной строке дано целое число k , $1 \leq k \leq 10^9$.

Формат выходных данных

Выведите одно целое число – количество хороших задач по модулю $10^9 + 7$. Если хороших задач бесконечное количество, выведите -1 .

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 | 3 |
| 260 | 15 |

Задача E. Выпуклый анализ

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

Гарри Джеймс Поттер-Эванс-Веррес — юный вундеркинд. В 11 лет Гарри знаком с квантовой механикой, когнитивной психологией, теорией вероятностей и другими вещами. И вот сейчас для Гарри пришло время изучить ещё один предмет — выпуклый анализ.

Начать, конечно же, стоит с азов. Фигура называется выпуклой если вместе с любой парой точек она также содержит отрезок их соединяющий. Простейшим примером выпуклой фигуры может служить выпуклый многоугольник. Гарри хочет вырезать такую фигуру из бумаги и исследовать (найти опорную функцию, сопряжённое множество и другие важные характеристики выпуклого множества).

Итак, у мальчика есть лист бумаги в клеточку размера $n \times m$ клеток. Он хочет отметить вершины многоугольника в целочисленных точках сетки, а затем вырезать его. При этом Гарри хочет, чтобы любые две соседние стороны многоугольника не лежали на одной прямой, а его площадь была равна S .

Формат входных данных

Даны два целых положительных числа n и m ($1 \leq n, m \leq 10^5$), а также вещественное число S ($0 < S \leq 10^9$) с не более, чем шестью знаками после запятой.

Формат выходных данных

Если построить многоугольник, удовлетворяющий данным условиям невозможно, выведите -1 . Иначе выведите число k — количество вершин многоугольника ($3 \leq k \leq 10^5$). В следующих k строках выведите k пар чисел x_i, y_i — координаты вершин искомого многоугольника ($0 \leq x_i \leq n$, $0 \leq y_i \leq m$) в порядке их обхода по часовой стрелке.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|--|
| 10 10 81 | 5 0 0 0 7 9 9 10 9 10 0 |
| 1 2 0.5 | 3 0 0 0 1 1 1 |

Задача F. Строковое дерево

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

Робот потерялся в большом городе. Город представляет собой совокупность n перекрёстков и двусторонних дорог, соединяющих некоторые из пары них. Всего в городе $n - 1$ дорога, гарантируется, что от любого перекрёстка можно добраться до любого другого, передвигаясь только по дорогам. Каждая дорога окрашена в один из 26-ти цветов, обозначаемых маленькими латинскими буквами (от 'a' до 'z').

Робот бродит по городу уже довольно давно и успел собрать некоторую статистику. А именно, пусть u, v – два различных перекрёстка в городе. Робот знает, что в данных ограничениях между этими двумя перекрёстками существует ровно один простой путь (то есть не проходящий ни по каким вершинам более чем единожды) по дорогам. Назовём *меткой* этого пути последовательность цветов дорог в порядке следования от u до v , записанную без каких-либо разделителей как строка латинских букв. Робот перебрал все пары различных перекрёстков и выписал для каждой из них метку соответствующего пути.

Заметим, что если известна метка s пути от u до v , то метка обратного пути есть просто развёрнутая строка s . Поэтому из всех $n(n - 1)$ упорядоченных пар робот сообщает Вам только $\frac{n(n-1)}{2}$, исключая обратные пути.

Вам нужно помочь роботу и составить подробный план города, описав все имеющиеся в городе дороги.

Формат входных данных

В первой строке задано число n ($2 \leq n \leq 100$) – количество перекрёстков в городе. В следующих $\frac{n(n-1)}{2}$ строках содержится информация о метках некоторых путей. В i -ой из этих строк содержатся через пробел два числа u_i, v_i ($1 \leq u_i, v_i \leq n$) и строка s_i , состоящая только из маленьких латинских букв, задающая метку ориентированного пути от u_i до v_i .

Гарантируется, что не существует i и j таких, что $u_i = v_j$ и $v_i = u_j$, то есть каждая пара задана ровно единожды. Также гарантируется, что робот Вас нигде не обманул, и требуемый план города существует.

Формат выходных данных

Выведите $n - 1$ строку, в j -ой из которых содержится описание j -ой дороги в формате $a_j b_j c_j$, где a_j, b_j – концы текущей дороги, а c_j – цвет этой дороги (маленькая латинская буква). Порядок выводимых дорог не имеет значения, концы дороги можно выводить в произвольном порядке.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------------------|-------------------|
| 3 1 2 a 1 3 b 2 3 ab | 1 2 a 1 3 b |
| 2 2 1 a | 2 1 a |

Задача G. Нападение аквилиан

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

После многих лет полета флот аквилиан достиг Земли. Они зависли над поверхностью и готовятся выпустить вирус. Последняя надежда человечества - сделать роботов, которые будут наблюдать за поверхностью Земли и ликвидировать все очаги заражения. Единственная сложность - роботы не должны быть обнаружены аквилианами, для этого они должны избегать линий сканирования кораблей враждебного флота. Вам нужно рассчитать, какое минимальное количество роботов необходимо сделать, чтобы вся поверхность Земли была защищена. Как всем известно, поверхность Земли представляет собой бесконечную плоскость, а линия сканирования одного корабля представляет собой параболу вида $y = (x - a_i)^2 + b_i$.

Формат входных данных

На первой строчке написано одно натуральное число $1 \leq n \leq 1000$ - количество кораблей флота аквилиан. Далее, на следующих n строках указано по два целых числа a_i, b_i , разделенных пробелом, $|a_i| \leq 10^9, |b_i| \leq 10^9$.

Формат выходных данных

Выведите единственное натуральное число - количество участков, на которые разбивают сканирующие линии поверхность Земли.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------------------|-------------------|
| 4 0 0 1 1 0 1 2 1 | 10 |
| 3 2 4 3 4 2 4 | 4 |

Задача N. Затопления в городе

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

Город Болотный часто подвергается затоплениям. При этом в городе всего одна улица и все дома города расположены на одной стороне этой улицы. У каждого дома есть целочисленная высота a_i , $1 \leq a_i \leq 10^9$. При этом кроме домов на улице также расположены источники воды, из-за которых возникают затопления. Высота каждого источника воды равна нулю.

Жители города не любят, когда на их части улицы появляется вода. В связи с этим жители шлют в мэрию города запросы – появится ли вода на участке улицы от L до R , если вода из всех источников города поднимется на уровень H . При этом, если на участке улицы от L до R есть хотя бы один источник воды, то ответ всегда будет “Yes” даже при $H = 0$.

Если вода поднялась на уровень H , то из каждого источника вода распространяется в каждую из сторон и затопляет дома до тех пор, пока не встретит на своем пути дом высотой больше или равной H .

Формат входных данных

В первой строке через пробел даны два целых числа n и m – длина улицы и количество запросов жителей, $1 \leq n, m \leq 10^6$.

Во второй строке через пробел даны n целых чисел a_1, a_2, \dots, a_n – высоты домов и источников, $0 \leq a_i \leq 10^9$. Если $a_i = 0$, значит это источник воды, иначе это дом высоты a_i .

В следующих m строках через пробел даны три целых числа L, R, H – параметры запроса, $1 \leq L \leq R \leq n$, $0 \leq H \leq 10^9$.

Формат выходных данных

Требуется вывести m строк, содержащих “Yes” или “No” без кавычек – ответ на запрос жителей.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6 7 | Yes |
| 0 2 3 1 4 0 | No |
| 1 1 0 | No |
| 4 4 2 | Yes |
| 2 5 2 | No |
| 2 5 3 | Yes |
| 3 5 3 | No |
| 3 5 4 | |
| 5 5 4 | |

Задача I. Разрезание на параллелограммы

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

Вам дан выпуклый многоугольник. Никакие две вершины многоугольника не совпадают. Никакие три вершины многоугольника не лежат на одной прямой. Требуется определить, можно ли разрезать заданный многоугольник на 2017 параллелограммов.

Формат входных данных

В первой строке записано целое число $3 \leq n \leq 100$.

В следующих n строках даны координаты вершин многоугольника в порядке обхода по часовой стрелке. В $(i + 1)$ -й строке даны два числа $-10^9 \leq x_i, y_i \leq 10^9$ – координаты i -й вершины.

Формат выходных данных

В единственной строке выведите “yes” без кавычек, если заданный многоугольник можно разрезать на 2017 параллелограммов. В противном случае выведите “no” без кавычек.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------------------|-------------------|
| 4 0 0 0 1 1 1 1 0 | yes |
| 3 0 0 0 1 1 0 | no |

Задача J. Петя и метро

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

Волею судеб, однажды Петя попал в город, где открылся новый метрополитен - „МетроДрево“, как его называют в городе. Причина такого названия заключается в следующих свойствах:

1. Метро состоит из n станций и $n - 1$ туннелей, каждый из которых соединяет ровно две станции;
2. По каждому туннелю курсируют поезда в обе стороны;
3. К каждой станции может подходить неограниченное число туннелей;
4. От каждой станции можно доехать до любой станции (возможно, с пересадками).

Петя всегда очень любил метро. Воспользовавшись тем, что поезда ездят быстро, а станций (пока) не очень много, юный путешественник перебрал все неупорядоченные пары станций и для каждой пары совершил поездку от одной станции к другой, проезжая при этом минимальное число туннелей. Оказалось, что в совокупности Петя проехал ровно m туннелей.

Петя так увлекся поездками, что, собираясь домой, забыл в гостинице схему метро! Вернувшись домой, Петя решил восстановить схему, используя числа n и m и считая, что все станции занумерованы от 1 до n . Помогите ему!

Формат входных данных

В единственной строке записаны два числа n , $2 \leq n \leq 20$, и m , $1 \leq m \leq 10000$.

Формат выходных данных

Если Петя ошибся (а такое, к сожалению, возможно), и удовлетворяющей условию задачи схемы не существует, выведите в единственной строке слово NO. Иначе, выведите в первой строке слово YES, после чего в $n - 1$ последующих строчках выведите по паре натуральных чисел от 1 до n - номеров станций, соединяемых очередным туннелем. Туннели можно выводить в любом порядке, равно как и станции для каждого туннеля. Если таких схем несколько, выведите любую.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 4 | YES 1 2 2 3 |
| 3 5 | NO |