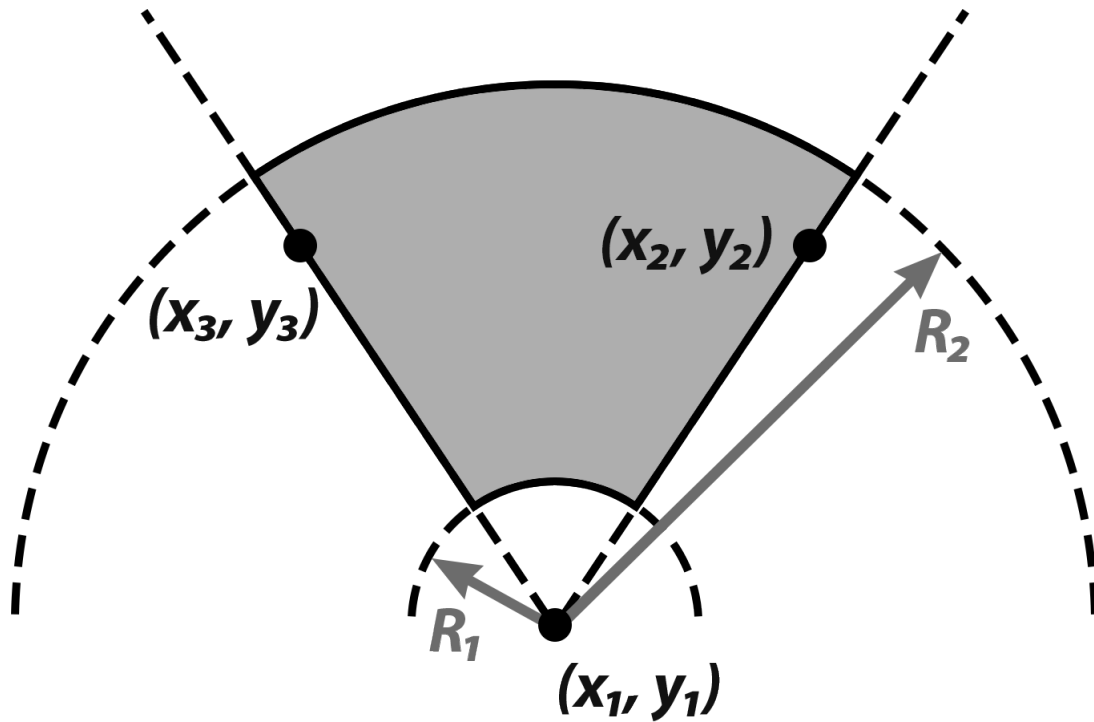


Задача А. Строгая отчётность

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

Как известно, колонизация планеты RT-92015-YS проходит успешно. Сегодня завершено размещение антенн глобальной распределённой сети. Антенна обеспечивает связь на площади поверхности, ограниченной двумя окружностями с радиусами R_1 и R_2 ($R_1 < R_2$) с центрами в точке расположения антенны, а также двумя прямыми, пересекающимися в этой же точке (см. рисунок).



Рабочий день завершён, и Евгению Землянину осталось лишь внести в отчёты площади, покрываемые сигналом каждой антенны. Однако заглянув в формуляр, Евгений обнаружил замысловатую графу, соответствующую параметру площади антенны:

$$A * Pi * arccos(B)$$

Ниже давалось пояснение: « Pi » означает число пи, « $arccos(B)$ » — функцию арккосинуса от числа B , принимающую значения на отрезке $[-1; 1]$ и возвращающую значения на отрезке $[0; 360]$. Вместо A и B может быть записано либо **целое** число, либо строка вида « C/D » (без кавычек), обозначающая число $\frac{C}{D}$, где C — **целое**, а D — **натуральное**. Дробь $\frac{C}{D}$ должна быть **несокращаемая**, то есть должно выполняться условие $gcd(C, D) = 1$, где gcd — наибольший общий делитель. Если строка « C/D » заменяет A , она должна быть дополнительно окружена круглыми скобками.

Евгений привык заполнять странные отчёты, ведь их форма определяется Бюро аналитики и обусловлена удобством формирования статистики. Но вычислять все необходимые значения довольно трудоёмко. Вспомнив о Вас, Евгений обратился к Вам за помощью. Напишите программу, которая по точке расположения антенны, двум точкам на граничных прямых и радиусам окружностей определяет площадь покрытия антенны в указанной форме.

Формат входных данных

Первая строка содержит натуральное число T ($1 \leq T \leq 10^5$) — количество наборов входных данных.

Каждый набор состоит из 4 строк. Первые три строки описывают целочисленные координаты трёх точек x_i и y_i , разделённые пробелом ($-10^9 \leq x_i, y_i \leq 10^9$) (см. рисунок). Гарантируется, что

$y_2 = y_3$, $x_2 - x_1 = x_1 - x_3$ и $y_1 < y_2$. Последняя строка набора содержит радиусы R_1 и R_2 , разделённые пробелом ($0 \leq R_1 < R_2 \leq 10^9$). Гарантируется, что площадь, которую необходимо вычислить, всегда положительна.

Формат выходных данных

Выведите площадь фигуры, ограниченной двумя прямыми и двумя окружностями, отформатированную в соответствии с указанной формой. Обратите внимание: вам необходимо вывести **большой объём данных**, это может стать причиной превышения ограничения по времени. Возможно, потребуется воспользоваться методами ускорения вывода.

Система оценки

Подзадачи	Ограничения	Баллы	Необходимые подзадачи	Тип проверки
1	Угол между прямыми 90°	20	—	Каждый тест
2	$(-10^6 \leq x_i, y_i, R_1, R_2 \leq 10^6)$	70	1	Каждый тест
3	Нет ограничений	10	1, 2	Полная

Пример

стандартный ввод	стандартный вывод
3	(1/18)*Pi*arccos(0)
0 0	(7/40)*Pi*arccos(3/5)
2 2	(1/120)*Pi*arccos(9991/10009)
-2 2	
4 6	
1 1	
3 5	
-1 5	
1 8	
0 0	
3 100	
-3 100	
1 2	

Задача В. Ягодная поляна

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

Бабушка отправила Петю за ягодами. Он набрел на особенную полянку размера $N \times N$. Поскольку полянка особенная, собирать на ней ягоды можно только определенным способом. Находясь в любой из граничных точек, необходимо выбрать направление по диагонали и, двигаясь только в нем, собрать все ягоды, которые встретятся по пути. Собрать ягоды можно только по одной диагонали.

Рядом с полянкой есть карта, на которой указано, насколько вкусные ягоды располагаются в каждой ее точке.

Петя, конечно же, хочет принести бабушке самые вкусные ягоды, поэтому собирать их будет, ориентируясь на карту. Причем, даже если самой вкусной из всех окажется корзина ягод с отрицательным или равным нулю итоговым значением, Петя все равно соберет ее.

Вам необходимо ответить, насколько вкусной будет ягодная корзина, которую Петя принесет бабушке. Итоговый вкус корзины равен сумме вкусов каждой собранной ягоды.

Формат входных данных

Первая строка содержит одно целое число N ($1 \leq N \leq 100$) — размер полянки.

В следующих N строках записано по N чисел от -100 до 100 , которые описывают полянку на карте. Каждая цифра описывает вкус ягод, находящихся в каждой точке полянки.

Формат выходных данных

Выведите вкус ягодной корзины, которую Петя принесет бабушке.

Система оценки

Подзадачи	Ограничения	Баллы	Необходимые подзадачи	Тип проверки
1	$N \leq 100$	100	—	Каждый тест

Пример

стандартный ввод	стандартный вывод
4 14 3 -7 1 2 -8 6 58 3 0 -2 -17 -45 -15 1 8	51

Задача С. Гномы и динамит

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

Третий подряд Гномов-шахтеров получил из распределителя указание помочь Второму подряду собрать алмазы в шахте горы Даймонд.

Приехав в указанное место, они не нашли гномов Второго подряда. Оказалось, что они уехали днем раньше. Зайдя в шахту, Гномы обнаружили карту шахты с указанием расположения алмазов и записку:

«Уважаемые Товарищи Гномы трудового Третьего подряда!

Горная порода не поддавалась нашим киркам! Поэтому Мы — Гномы Второго подряда — заложили в шахту разработанный нами динамит «КАБУМ-12»! Динамит получился специфичный, пока его не активируют, он не взрывается. Динамит очень мощный — чтобы взорвать горную породу и сохранить алмаз, необходим только 1 взрыв, но при наложении действия одного динамита на действие другого взрыв уничтожает ВСЕ алмазы на данной области.

Список областей действия каждой шашки динамита прилагаем ниже!

Удачного Сбора Алмазов!

Второй подряд!»

Вам, как главному Гному-инженеру, необходимо рассчитать максимальное количество алмазов, которые можно собрать, если взорвать весь заложенный в шахте «КАБУМ-12»!

Формат входных данных

Первая строка содержит два целых числа N и M ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^5$) — количество Количество ячеек в шахте и количество шашек динамита соответственно.

Вторая строка содержит N целых чисел X_i — количество алмазов в i -го секторе ($1 \leq X_i \leq 727$, $1 \leq i \leq N$).

Далее следует M строк, которые содержат по два числа L_j и R_j — границы действия j -й шашки динамита ($1 \leq j \leq M$, $1 \leq L_j \leq N$, $L_j \leq R_j \leq N$).

Формат выходных данных

Выведите одно число — максимальное количество алмазов, которые можно собрать.

Система оценки

Подзадачи	Ограничения	Баллы	Необходимые подзадачи	Тип проверки
1	$N, M \leq 10^3$	15	—	Каждый тест
2	$N, M \leq 10^4$	25	—	Каждый тест
3	$N, M \leq 10^5$	60	—	Полная

Примеры

стандартный ввод	стандартный вывод
5 3 1 0 3 7 2 1 3 2 4 4 5	3
9 5 5 9 2 0 0 7 1 0 3 1 9 2 5 2 7 4 7 5 8	8

Задача D. Проблемы необычного фермера

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Однажды один программист Александр настолько устал от городской жизни, от непостоянного графика, отсутствия личной жизни, постоянных дедлайнов и злости на заказчиков, которые сами не знают что хотят получить, что решил бросить свою работу и перебраться в деревню, чтобы стать фермером. Он купил себе небольшой участок земли вместе с домом, благо денег у него хватало, ведь программистом он был отличным. На этом участке земли было прямоугольное поле, пригодное для посева. Это поле можно условно разделить на $N * M$ прямоугольных участков. А каждый участок поля характеризуется коэффициентом плодородия $A_{i,j}$ ($1 \leq i \leq N, 1 \leq j \leq M$).

И вот после того как он немного освоился в деревне пришла ему пора заняться сельским хозяйством. Для того, чтобы вырастить хороший урожай он решил выбрать такую прямоугольную область поля, чтобы суммарный коэффициент плодородия этой области был максимально возможным. То есть он хочет выбрать в матрице A такую подматрицу, что бы сумма её элементов была максимальной.

Александр понимает, что эта задача сложная и решить её быстро можно только прибегнув к программированию. Но так как он пообещал себе, что больше не притронется ни к одной IDE и не напишет ни единой строчки кода, он просит вас помочь написать программу, которая по заданным значениям коэффициентов плодородия найдет координаты области с самым большим суммарным коэффициентом плодородия.

Формат входных данных

Первая строка содержит два натуральных числа N и M ($1 \leq N \leq 400, 1 \leq M \leq 10^5$) — размеры поля.

i -я из следующих N строк содержит M целых чисел $A_{i,1}, A_{i,2}, \dots, A_{i,M}$ ($-10^5 \leq A_{i,j} \leq 10^5$) — коэффициент плодородия участка поля с координатами i и j . Гарантируется, что хотя бы один участок имеет положительный коэффициент.

Формат выходных данных

В первой строке выведите одно целое число — максимальный суммарный коэффициент плодородия области поля.

Во второй строке выведите два целых числа через пробел X_1, Y_1 ($1 \leq X_1 \leq N, 1 \leq Y_1 \leq M$) — координаты левого верхнего участка области.

В третьей строке выведите два целых числа через пробел X_2, Y_2 ($1 \leq X_2 \leq N, 1 \leq Y_2 \leq M$) — координаты правого нижнего участка области.

Причем $X_1 \leq X_2$ и $Y_1 \leq Y_2$.

Система оценки

Подзадачи	Ограничения	Баллы	Необходимые подзадачи	Тип проверки
1	$N = 1, 1 \leq M \leq 5000$	15	—	Полная
2	$N = 1, 15000 \leq M \leq 100000$	35	1	Каждый тест
3	$10 \leq N \leq 75, 10 \leq M \leq 75$	15	—	Полная
4	$100 \leq N \leq 400, 100 \leq M \leq 400$	35	3	Каждый тест

Примеры

стандартный ввод	стандартный вывод
1 10 -6 1 4 2 -7 -14 3 -1 6 8	16 1 7 1 10
2 10 15 -6 11 10 -15 0 14 8 1 7 7 1 8 14 0 -15 10 11 -6 15	90 1 1 2 10

Задача Е. Не римские операции

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

На уроке информатики учитель объясняла ученикам как работают римские цифры:

Всего существует 7 чисел - $I(1)$, $V(5)$, $X(10)$, $L(50)$, $C(100)$, $D(500)$, $M(1000)$. Все целые числа от 1 до 3999 записываются с помощью приведенных выше цифр. При записи римских чисел необходимо соблюдать правила.

Если большая цифра стоит перед меньшей, они складываются (до 3-х цифр): $VI = 5 + 1 = 6$; $VIII = 10 + 3 = 8$;

Если меньшая цифра стоит перед большей (в этом случае она не может повторяться), то меньшая вычитается из большей; вычитаться могут только цифры, обозначающие 1 или степени 10; уменьшаемым может быть только цифра, ближайшая в числовом ряду к вычитаемой $IV = 5 - 1 = 4$; $IX = 10 - 1 = 9$;

Цифры V, L, D не могут повторяться; цифры I, X, C, M могут повторяться не более трех раз подряд: $VIII = 8$; $LXXX = 80$;

Михаилу не понравилась эта система счисления, потому что она неудобно хранилась в компьютере, но ему стало интересно что получится, если последовательно применять к римским числам интересные операции. Для его экспериментов учитель написала программу которая дает последовательность римских чисел. Чтобы было интереснее, Михаил придумал 3 группы. **Элегантные**, **обычные** и **удивительные**. Числа попадают в них по максимальному количеству повторяющихся символов. Для **элегантной** не более 1 повторяющегося символа, для **обычной** 2 или 3, для **удивительной** 4.

Михаил решил объединить все числа в одно следующим образом. Взять первое число как начальный результат, после этого использовать все числа по очереди, начиная со второго. В зависимости от группы, к которому принадлежит это число произвести между ним и текущим результатом определенные операции. Представим числа как сумму степеней двоек (как в двоичной системе счисления).

1. Если попало число из первой группы, то результатом будет сумма только тех степеней двоек, которые присутствуют в обоих числах.
2. Если попало число из второй группы, то результатом будет сумма только тех степеней двоек, которые присутствуют в одном числе, но отсутствуют в другом.
3. Если попало число из третьей группы, то к исходному результату прибавляется количество различных степеней двоек в текущем числе.

Получившееся число преобразуется обратно в римскую систему счисления.

Взяв первое число как начальное, Михаил должен был распределять последующие числа на группы и применять к ним соответствующие операции, но он устал, потому что всю ночь смотрел видео по хеш-таблицам. Помогите Михаилу решить эту задачу.

Формат входных данных

Первая строка содержит целое число T ($1 \leq T \leq 50$) — количество тестов.

Далее следуют T пар строк. Первая строка целое число R ($5 \leq R \leq 200$) — количество чисел в римской системе счисления.

Вторая строка содержит R строк, разделенных пробелом — римские числа (минимальное число 1, максимальное 3999).

Формат выходных данных

Вывод итогового числа в римской системе счисления для каждого набора по модулю 4000. (Если получится 0, то выведите N).

Система оценки

Подзадачи	Ограничения	Баллы	Необходимые подзадачи	Тип проверки
1	$1 \leq T \leq 50, 1 \leq R \leq 200$	100	—	Каждый тест

Пример

стандартный ввод	стандартный вывод
1 4 X I II LXXXIX	VI

Замечание

В первом примере римское число X мы переводим в $2^3 + 2^1$ так как его двоичное представление это 1010.

Теперь нам нужно посмотреть на следующее число. I мы представляем как $2^0(1)$. Так как в числе I одинаковые римские цифры встречается максимум один раз, то мы сохраняем только повторяющиеся двойки, которые встречаются в том и в другом числе. Их у нас нет, поэтому после этого этапа у нас остается 0.

Следующее число $II(10)$. В нем две одинаковые римские цифры. Смотрим на не повторяющиеся двойки — 2^1 . В 0 нет вообще цифр, поэтому после этого действия результат будет 2.

Последнее число $LXXXIX(1011001)$. В нем четыре одинаковые римские цифры. Поэтому складываем исходное число II с количеством различных степеней двоек. $2^6 + 2^4 + 2^3 + 2^0$ $2 + 4 = 6 = VI$.